



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNOLOGIÍ

ÚSTAV AUTOMATIZACE A MĚŘICÍ TECHNIKY

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF CONTROL AND INSTRUMENTATION

ADAPTACE PARAMETRŮ VE FUZZY SYSTÉMECH

ADAPTATION OF PARAMETERS IN FUZZY SYSTEMS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Ing. MILOSLAV FIC

VEDOUCÍ PRÁCE
SUPERVISOR

doc. Ing. VÁCLAV JIRSÍK, CSc.

BRNO 2015



**VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ**

**Fakulta elektrotechniky
a komunikačních technologií**

Ústav automatizace a měřicí techniky

Diplomová práce

magisterský navazující studijní obor
Kybernetika, automatizace a měření

Student: Ing. Miloslav Fic

ID: 120275

Ročník: 2

Akademický rok: 2014/2015

NÁZEV TÉMATU:

Adaptace parametrů ve fuzzy systémech

POKYNY PRO VYPRACOVÁNÍ:

1. Seznamte se s problematikou adaptace parametrů ve fuzzy systémech.
2. Zaměřte se na adaptace s využitím umělých neuronových sítí.
3. Definujte typ fuzzy systému a vyberte vhodnou neuronovou síť pro adaptaci parametrů fuzzy systému.
4. Pomocí vybrané umělé neuronové sítě proveďte adaptaci parametrů definovaného fuzzy systému.
5. Dosažené výsledky zhodnoťte.

DOPORUČENÁ LITERATURA:

P.Jura: Základy fuzzy logiky pro řízení a modelování. VUTIM, Brno 2013, ISBN 80-214-2261-0

Termín zadání: 9.2.2015

Termín odevzdání: 3.8.2015

Vedoucí práce: doc. Ing. Václav Jirsík, CSc.

Konzultanti diplomové práce:

doc. Ing. Václav Jirsík, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

Abstrakt

Diplomová práce se zabývá problematikou adaptace parametrů fuzzy systémů a to především umělými neuronovými sítě. V rešeršní části je proveden rozbor dosavadních metod spojujících fuzzy systémy a umělé neuronové sítě včetně uvedení zdrojů mezi studentskými pracemi. V kapitole aplikace metod je věnován prostor ověření klasifikační schopnosti zvolené metody fuzzy-neuronové sítě v Kohonenovým učení a následně návrhu modelu fuzzy systému na bázi této metody.

Klíčová slova

Fuzzy-Neuronová síť, Kohonenova síť, FKCN, fuzzy systém, adaptace parametrů

Abstract

This Master's thesis deals with adaptation of fuzzy system parameters with main aim on artificial neural network. Current knowledge of methods connecting fuzzy systems and artificial neural networks is discussed in the search part of this work. The search in Student's works is discussed either. Chapter focused on methods application deals with classifying ability verification of the chosen fuzzy-neural network with Kohonen learning algorithm. Later the model of fuzzy system with parameters adaptation based on fuzzy-neural network with Kohonen learning algorithm is shown.

Keywords

Fuzzy-Neural Network, Kohonen Network, FKCN, fuzzy system, parameters adaptation.

Bibliografická citace:

FIC, M.: *Adaptace parametrů ve fuzzy systémech*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2015. 56 s. Vedoucí diplomové práce byl doc. Ing. Václav Jirsík, CSc.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma *Adaptace parametrů ve fuzzy systémech* jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne: **3. srpna 2015**

.....
podpis autora

Poděkování

Děkuji vedoucímu diplomové práce doc. Ing. Václavovi Jirsíkovi, CSc. za účinnou metodickou, pedagogickou a odbornou pomoc a další cenné rady při zpracování mé diplomové práce.

V Brně dne: **3. srpna 2015**

.....
podpis autora

Obsah

1	Úvod.....	8
2	Fuzzy Systémy.....	9
2.1	Funkce příslušnosti.....	9
2.2	Operace s fuzzy množinami	11
2.3	Fuzzy relace	12
2.4	Přibližné usuzování	13
2.5	Vlastnosti souboru pravidel.....	13
2.6	Základní schéma fuzzy systému.....	14
2.7	Typy fuzzy systémů	15
2.8	Aplikace fuzzy systémů	15
3	Adaptace parametrů fuzzy systému	16
3.1	Modely fuzzy systému s adaptací parametrů	16
4	Fuzzy-Neuronové sítě.....	17
4.1	ANFIS	20
4.2	Fuzzy-neuronové sítě s učením bez učitele.....	23
4.3	Kohonenova síť	23
4.3.1	Životní cyklus sítě.....	23
4.3.2	Doporučené nastavení Kohonenovy sítě.....	26
4.3.3	Aplikace Kohonenovy sítě.....	27
4.3.5	Modifikace základního algoritmu	28
4.3.6	Tvorba neuro-fuzzy klasifikátoru na Kohonenově síti	29
4.4	Fuzzy Kohonen Clustering Network.....	31
4.4.1	Modifikace FKCN	34
4.4.2	Aplikace FKCN	34
5	Aplikace zvolené metody.....	35
5.1	Cíle praktického nasazení vybraného algoritmu	35
5.2	Tvorba datových souborů.....	35
5.3	Učení algoritmu.....	38
5.3.1	Srovnání kvality třídění	42
5.4	Stavba fuzzy systému	48
6	Závěr	51
7	Literatura.....	52

1 ÚVOD

Fuzzy systémy jsou algoritmy s velmi širokým rozsahem aplikace. Jejich nespornou výhodou je schopnost zpracovávat neurčité informace a mohou pracovat nejen s kvantitativními proměnnými, ale také s kvalitativními. Aplikace fuzzy systémů se nachází jak v oblasti business managementu a ratingu, řízení a regulaci nebo modelování nelineárních funkcí, tak v lékařství např. při klasifikaci spánkových EEG a odhalování mikrosnátku.

Nastavení fuzzy systému je pro rozsáhlejší problémy bez přesnější znalosti vnitřní struktury obtížnější problém. Za tímto účelem byly vyvinuty algoritmy, které mají schopnost učit se a adaptovat parametry fuzzy systémů tak, aby dosáhly požadované přesnosti, např. aproximace nebo klasifikace.

V této práci je uvedena základní teorie fuzzy systémů v druhé kapitole.

Ve třetí kapitole navazuje vymezení parametrů vhodných k adaptaci a rozdělení algoritmů podle modelu učení.

Čtvrtá kapitola představuje řešební část v přehledem literárních zdrojů. V této kapitole jsou představeny fuzzy-neuronové sítě, představující pomezí mezi fuzzy systémy a neuronovými sítěmi. Především je zaměřena na řešební metod spojených se shlukovou analýzou, předně Kohonenova síť a její adaptace. Je zde také uveden dopředný algoritmus ANFIS a modifikace Kohonenovy sítě FKCN, v níže je uplatněno fuzzy učení na bázi fuzzy C-means. V kapitole 4.3.2 je rozbor doporučení, vyplývajících z předchozí analýzy Kohonenovy sítě zpracované ve zdroji [6].

Pátá kapitola uvádí algoritmus fuzzy-neuronové sítě včetně popisu jednotlivých funkcí a schématu vytvořeného programu v prostředí MATLAB. Protože zvolená fuzzy-neuronová síť je určena ke klasifikaci, je v rámci kapitoly provedeno srovnání pro 3 datové soubory s Kohonenovou sítí a FKCN. Ve všech sítích se uplatnila modifikace ve smyslu LVQ1 pro Kohonenovy sítě. V kapitole 5.4 je popsán jednoduchý fuzzy systém, který je na učení fuzzy-neuronové sítě postaven.

2 FUZZY SYSTÉMY

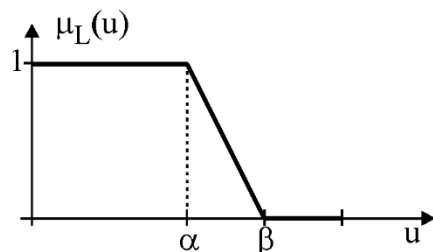
Fuzzy systém je systém založený na matematickém aparátu zvaném fuzzy logika. V automatizaci se setkáme s definicí systému: „Systém je definován jako účelově uspořádaná množina prvků a množina vazeb mezi nimi, s dynamickým chováním, které společně určují vlastnosti celku.“ Čerpáno ze zdroje [5]. Systémy pracují se vstupními, výstupními a stavovými proměnnými. Fuzzy systémy se od spojitých systémů liší vyjádřením proměnných. Ve fuzzy systémech jsou proměnné reprezentovány jazykovými proměnnými, tedy fuzzy množinami, a nenabývají reálných hodnot. Fuzzy množiny jsou odvozeny z teorie množin, tj. klasických (ostrých) množin.

Klasická množina je reprezentována souborem prvků. Počet prvků v množině může být konečný, nekonečný nebo nulový. Příkladem zápisu klasické množiny je $M = \{1,2,3,5,7,11\}$.

Fuzzy množiny nejsou svázány vyjádřením prvku přesnou hodnotou a pracují s vágními pojmy jako např. (teplý, vlažný, studený), (pomalý, rychlý, velmi rychlý). Každá fuzzy množina F je definován funkcí příslušnosti μ_F na celém universu U . Universum je speciálním případem množiny a obsahuje všechny prvky. [10]

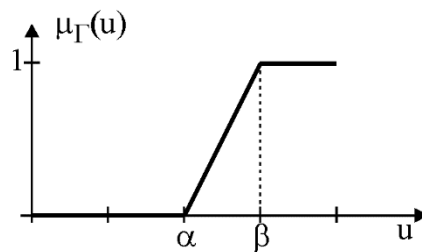
2.1 Funkce příslušnosti

Funkce příslušnosti mapuje universum na interval 0 až 1: $\mu_F: U \rightarrow \langle 0,1 \rangle$. Ze zápisu vyplývá, že prvek universa do fuzzy množiny F buď patří 1, nepatří 0, nebo patří částečně (0,1). Jakou měrou patří prvek do množiny je dáno tvarem funkce příslušnosti. V zásadě se využívá 6 tvarů funkce příslušnosti: L, Γ , Λ , Π , S (Zadeh), Π (Zadeh):



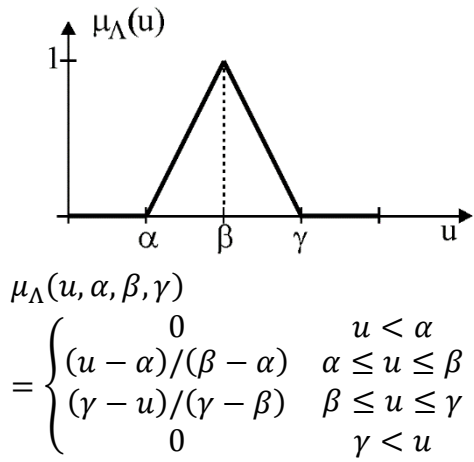
$$\mu_L(u, \alpha, \beta) = \begin{cases} 1 & u < \alpha \\ (\beta - u)/(\beta - \alpha) & \alpha \leq u \leq \beta \\ 0 & \beta < u \end{cases}$$

Obr. 1 Funkce příslušnosti L [6]

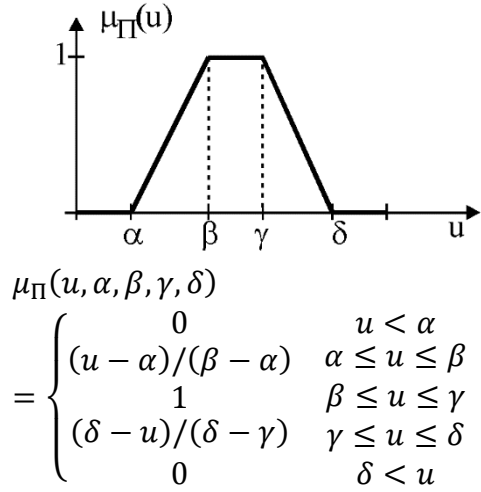


$$\mu_\Gamma(u, \alpha, \beta) = \begin{cases} 0 & u < \alpha \\ (u - \alpha)/(\beta - \alpha) & \alpha \leq u \leq \beta \\ 1 & \beta < u \end{cases}$$

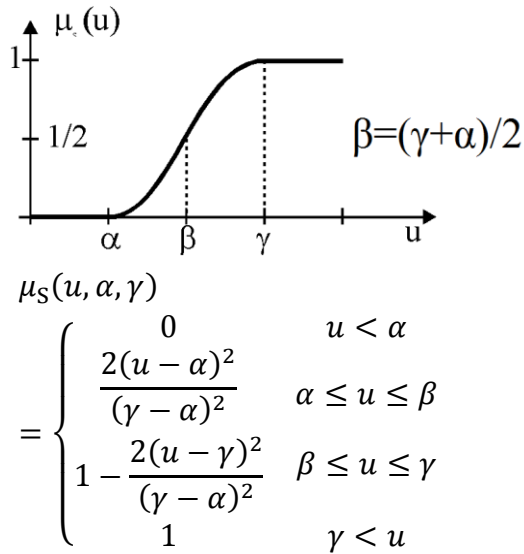
Obr. 2 Funkce příslušnosti Γ [6]



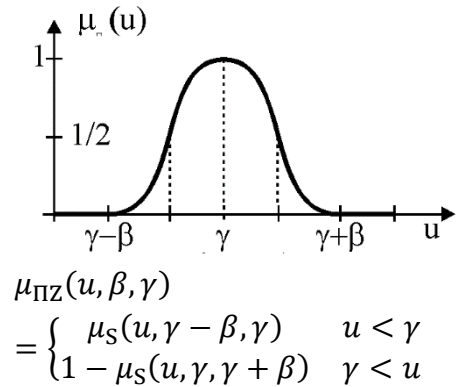
Obr. 3 Funkce příslušnosti Λ [10]



Obr. 4 Funkce příslušnosti Π [10][6]



Obr. 5 Funkce příslušnosti S (Zadeh)[10]

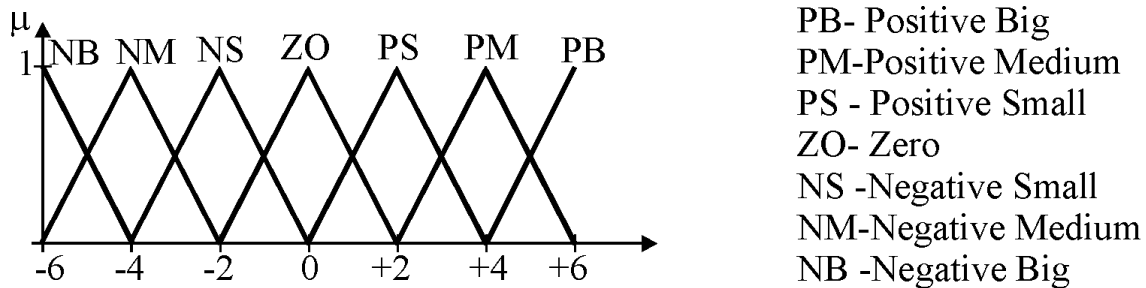


Obr. 6 Funkce příslušnosti Π (Zadeh) [10]

K jednoznačnému určení fuzzy množiny F je nezbytně nutná znalost prvku množiny $u \in U$ a jeho hodnoty funkce příslušnosti $\mu_F(u)$, zapsáno:

$$F = \{(u, \mu_F(u)) / u \in U\} \quad (1)$$

Obvyklé dělení při praktickém využití fuzzy logiky je na 3, 5, 7 množin značeno symbolicky (viz Obr. 7):



Obr. 7 Příklad 7 prvků fuzzy množiny na universu (-6,6) [10]

V případě menšího počtu množin se názvy prvků redukují, např. {Small, Zero, Big}

2.2 Operace s fuzzy množinami

Operace, které jsou známy u klasických množin nejsou u fuzzy množin tak jasně definovány jako např. *průnik* (2), *sjednocení* (3) a *doplňek* (4) fuzzy množin A a B.

Dle Zadeh (x je prvkem universa X) [10]:

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad (2)$$

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad (3)$$

$$\mu_{A^c}(x) = 1 - \mu_A(x) \quad (4)$$

Varianta 2:

$$\mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x) \quad (5)$$

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x) \quad (6)$$

Další operace:

Omezený
součet

$$A \oplus B = \int \min(1, \mu_A(x) + \mu_B(x)) / x \quad (7)$$

Omezený
rozdíl

$$A \ominus B = \int \max(0, \mu_A(x) - \mu_B(x)) / x \quad (8)$$

Součin

$$A \cdot B = \int \mu_A(x) \cdot \mu_B(x) / x \quad (9)$$

Součin

$$A^\alpha = \int (\mu_A(x))^\alpha / x \quad (10)$$

2.3 Fuzzy relace

Fuzzy relace vyjadřuje kartézský součin dvou fuzzy množin. Jedná se o základní kámen mapování vícerozměrných spojitých vstupů do fuzzy množin. Fuzzy relace je zapsána jako výraz [10]:

$$R = \int_{X \times Y} \mu_R(x, y) / (x, y), \quad (11)$$

kde x je prvek universa X ,
 y je prvek universa Y .

Operace s fuzzy relacemi

Stejně jako v případě fuzzy množin je pro fuzzy relace R a S definován průnik (12) a sjednocení (13).

$$\mu_{R \cap S}(x) = \min(\mu_R(x, y), \mu_S(x, y)) \quad (12)$$

$$\mu_{R \cup S}(x) = \max(\mu_R(x, y), \mu_S(x, y)) \quad (13)$$

Další operace:

$$\text{proj } R \text{ na } Y = \int_Y \max_{\forall x} \mu_R(x, y) / y \quad (14)$$

$$\text{ce}(A) = \int_{X \times Y} \mu_A(y) / (x, y) \quad (15)$$

$$B = A \circ R = \text{proj}(\text{ce}(A) \cap R) \text{ na } Y \quad (16)$$

Protože je průnik fuzzy množin definován dvěma způsoby, existují také rozdílné definice kompozice. Dle Zadeh:

$$\mu_B(y) = \max_{\forall x} \min(\mu_A(x), \mu_R(x, y)). \quad (17)$$

Alternativa:

$$\mu_B(y) = \max_{\forall x} (\mu_A(x) \cdot \mu_R(x, y)), \quad (18)$$

kde A, B jsou fuzzy množiny,
 R je relace,
 x, y jsou prvky definovány na universu X a Y .

2.4 Přibližné usuzování

Přibližné usuzování je hlavní mechanismus spojující lingvistické proměnné na vstupu s výstupy fuzzy systému. Základem přibližného usuzování je soubor pravidel. Každé pravidlo (tzv. inference) přibližného usuzování má tvar:

$$\text{IF } (u \text{ je } A) \text{ AND } (v \text{ je } B) \text{ THEN } (y \text{ je } C), \quad (19)$$

kde $(u \text{ je } A)$ a $(v \text{ je } B)$ a $(y \text{ je } C)$ jsou výrazy,

IF předchází předpokladu (antecedentu)

THEN předchází výsledku (konsekventu)

AND/OR uvádí způsob spojení předpokladů.

Pravidlo je reprezentováno *fuzzy implikací*: IF *fuzzy výraz* THEN *fuzzy výraz*. Fuzzy implikace lze vyjádřit jako Booleovkou, Larsen nebo Mamdani implikace (viz [10]). Vyhodnocení souboru pravidel může být provedeno dvěma způsoby:

- 1) Kompoziční způsob – Nejprve jsou sjednoceny fuzzy relace a následně se provede fuzzy implikace pro předloženou vstupní proměnnou na sjednocení fuzzy relací.
- 2) Individuální způsob – U individuálního zpracování se každé pravidlo vyhodnotí pro předloženou vstupní proměnnou zvlášť a sjednotí se výstupy jednotlivých pravidel.

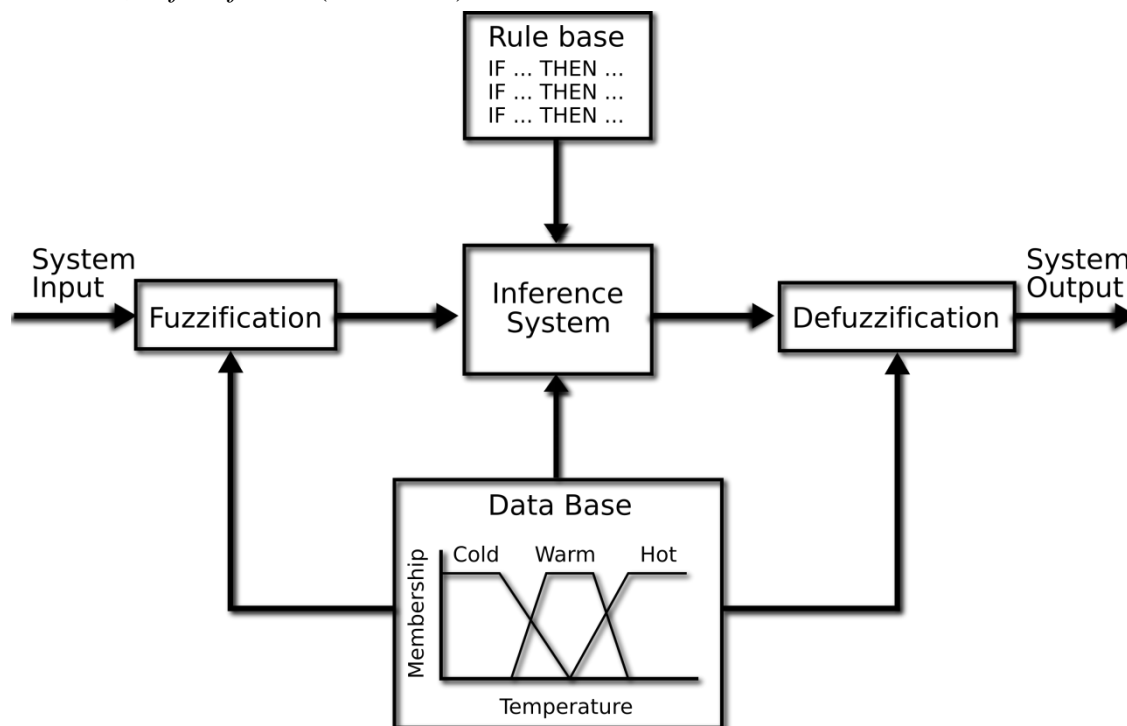
2.5 Vlastnosti souboru pravidel

Na soubor pravidel fuzzy systému jsou kladeny následující požadavky:

- 1) Úplnost souboru pravidel – Pro každý prvek vstupního prostoru existuje alespoň jedna fuzzy množina, jejíž funkce příslušnosti nabývá hodnoty větší než 0.
- 2) Konzistence souboru pravidel – V souboru pravidel existuje právě jedno pravidlo s danou variací antecedentů a pouze jedním konsekventem. Existence více než jednoho pravidla s danou variací antecedentů vede k nekonzistenci pravidel.
- 3) Spojitost souboru pravidel – Soubor pravidel je spojitý, jestliže pravidla se sousedními antecedenty mají také sousední konsekventy, tedy nenulový průnik konsekventů.
- 4) Interakce souboru pravidel – Výsledek vyhodnocení kompozitním a individuálním způsobem se liší. Potom je soubor pravidel interaktivní. [10]

2.6 Základní schéma fuzzy systému

Fuzzy systém obsahuje moduly *fuzzifikace*, *báze pravidel*, *báze dat*, *inference mechanismus*, *defuzzifikace* (viz Obr. 8)



Obr. 8 Obecné schéma fuzzy systému [11]

Fuzzifikace – proces převedení spojité proměnné na lingvistickou proměnnou a míru příslušnosti k příslušné fuzzy množině.

Báze pravidel – jedná se o soubor pravidel zachycující inference mezi antecedenty a konsekventem. Báze pravidel musí splňovat vlastnosti dle kap. 2.5.

Báze dat – obsahuje informace o funkcích příslušnosti fuzzy množin a o všech universech, se kterými se ve fuzzy systému pracuje včetně normalizačních a denormalizačních měřítek. Dále obsahuje informace o defuzzifikační metodě.

Inference mechanismus – provádí vyhodnocení vstupních proměnných vycházejících z fuzzifikace na základě báze pravidel formované za pomoci báze dat. Výstupem inference mechanismu je ohodnocení jednotlivých pravidel a konjunkcí hodnocení.

Defuzzifikace – představuje mechanismus výpočtu spojité výstupní proměnné z hodnocení inference mechanismem. Metody defuzzifikace jsou např. metoda těžiště CoG, metoda středu součtů CoS, metoda průměru středů CAM, metoda prvního maxima FoM, metoda středu maxima MoM. [10]

2.7 Typy fuzzy systémů

Rozlišují se základní dva typy fuzzy systémů. Jedním je *Mamdani* systém a druhým *Takagi-Sugeno*. Zásadním rozdílem mezi těmito druhy je tvar konsekventu.

- Mamdani

Konsekvent inference je ve tvaru lingvistických proměnných a nezbytně musí být součástí fuzzy systému defuzzifikační modul, je-li vyžadován výstup ve tvaru výstupní spojitě hodnoty.

- Takagi-Sugeno

Konsekvent inference je ve tvaru funkce vstupních proměnných a výstupem této funkce je přímo spojitá reálná hodnota. Následně se provede vážený součet a denormalizace výstupu. [10]

2.8 Aplikace fuzzy systémů

Fuzzy systémy se úspěšně používají při modelování lineárních i nelineárních systémů a řízení soustav. Fuzzy systémy vykazují schopnost predikce a jsou často využívány pro účely klasifikace. V případě klasifikace nám fuzzy systém poskytují nejenom informaci predikované příslušnosti ke třídě ale také míru, jakou k třídě náleží. Tato hodnota přímo vychází ze zvolených funkcí příslušnosti. Největší výhodou fuzzy systémů je *interpretovatelnost, modulárnost a transparentnost* báze pravidel a báze dat.

Fundamentální tvrzení vymezující aplikaci fuzzy systémů zní: „Libovolnou spojitou funkcí, definovanou na kompaktní množině, lze s libovolnou přesností aproximovat aditivním fuzzy systémem.“ Vychází ze zdroje [11].

3 ADAPTACE PARAMETRŮ FUZZY SYSTÉMU

Fuzzy systém je komplexní systém, ve kterém se vyskytuje velké množství parametrů. Parametry v bázi dat tvoří základní předmět adaptačních mechanismů. Při návrhu fuzzy systému nastavujeme hlavní parametry, jakými je počet vstupních proměnných a na ně navazující počet termů jednotlivých lingvistických proměnných. Term je jazykový výraz, kterého může proměnná nabývat. Zmíněné parametry jsou však velmi základní a proto se k jejich adaptaci přistupuje u pokročilejších metod. Změnou základních parametrů je míněna změna

- vztahů fuzzy inference,
- počtu vstupních proměnných,
- souboru fuzzy pravidel,
- tvaru funkcí příslušnosti k jednotlivým termům.

Vhodnějšími a častěji využívanými parametry k adaptaci jsou parametry funkcí příslušnosti (viz kap.2.1), příp. počtu termů. Mezi adaptované parametry patří α , β , γ i δ . Způsob, jakým se parametry adaptují, souvisí dále se zvoleným modelem adaptivního systému. V zásadě rozlišujeme mechanismy adaptace:

- metodou zpětného šíření chyby,
- metodou nejmenších čtverců,
- genetickými algoritmy,
- neuronovými sítěmi,
- shluková analýza.

3.1 Modely fuzzy systému s adaptací parametrů

Adaptace parametrů je možná dle libovolného algoritmu, který vykazuje konvergenci k minimu chybové funkce. Mezi často využívané algoritmy patří:

- 1) Metody nejmenších čtverců pro pokrytí fuzzy bazovými funkcemi
- 2) Fuzzy-neuronová síť s učením:
 - s učitelem (Algoritmus zpětného šíření chyby)
 - bez učitele (Samoorganizující síť)
 - s radiální bazovou funkcí

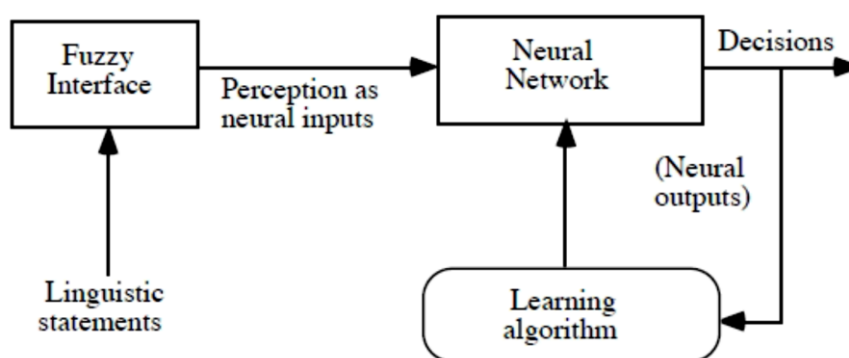
4 FUZZY-NEURONOVÉ SÍTĚ

Motivace spojení těchto dvou matematických prostředků je dána vlastnostmi každého z nich.

Fuzzy systémy jsou obecně uznávané v modelování nelineárních systémů a regulaci. Jejich vnitřní dynamika je jasně vyjádřena v bázi pravidel a bázi dat. Tato pravidla stejně jako data musí být do systému vložena při jeho vytvoření expertem. Během sestavení obou bází se klade důraz na minimalizaci odchylky skutečného výstupu fuzzy systému a žádaného výstupu, např. výstup modelovaného systému nebo žádaná akční veličina fuzzy regulátoru. V případě jednoduchých modelů není čas potřebný k sestavení bází tak zásadní, aby byl nutně expert nahrazen automatickým algoritmem. S narůstající komplexitou řešeného problému je tvorba bází kritickým prvkem v procesu nasazení fuzzy systému.

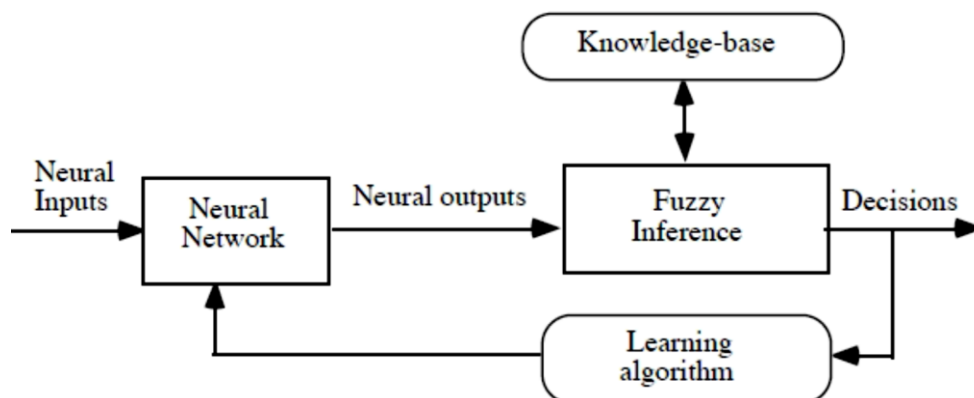
Neuronové sítě vystupují jako algoritmy se schopností adaptace na předložený systém. Jejich vnitřní dynamika je ovšem neurčitá a nelze ji vyjádřit prostým seznamem pravidel, které by bylo snadné ověřit, tedy jejich aplikace v průmyslové praxi je omezená. Spojení neuronových sítí a fuzzy systémů je nasnadě. Rozlišujeme modely [7]:

1. Fuzzy inferenční mechanismus pracuje jako překladač neostrých jazykových výrazů, které dále slouží jako vstupy neuronové sítě. Neuronová síť se adaptuje a zpřesňuje svůj výstup při žádané aplikaci.



Obr. 9 Fuzzy inferenční mechanismus na vstupu neuronové sítě. [7]

2. Neuronová síť slouží jako řídicí prvek vstupů fuzzy systému. Neuronová síť se učí na výstupní chybu celku, čímž předzpracovává vstupy fuzzy systém a nepřímo tak mění fuzzy usuzování celku.



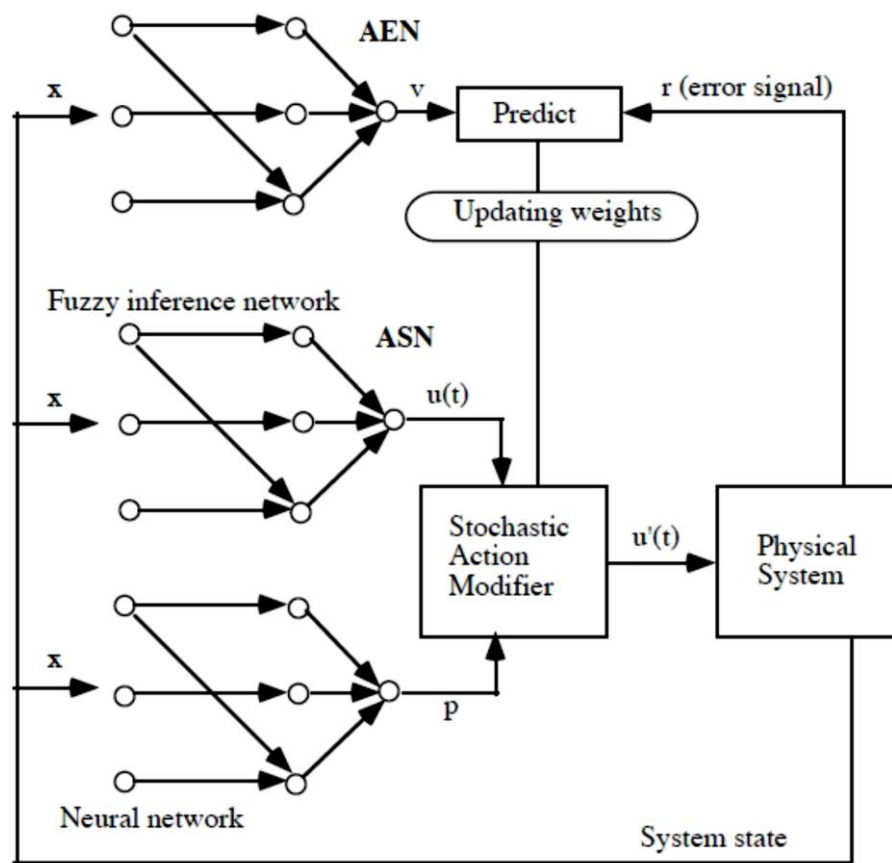
Obr. 10 Neuronová síť předzpracovávající vstupy fuzzy systému. [7]

Oba předchozí modely ukazují sériové spojení fuzzy systému a neuronové sítě.

3. *Berenjiho ARIC* (Approximate Reasoning Based Intelligent Control – česky: inteligentní řízení na základě přibližného usuzování). Jedná se o neuronovou síť modelující fuzzy systém. Model je založen na paralelním spojení dopředné neuronové sítě, neuronové sítě pro výpočet akčního stavu (AEN - Action-State Evaluation Network) a neuronové sítě pro výběr akce (ASN - Action Selection Network). Schéma Berenji's ARIC je na Obr. 11.

ASN je spojení dvou vícevrstevných sítí, kde první, modelující fuzzy inferenční mechanismus, na výstupu předkládá doporučenou hodnotu výstupu $u(t)$ pro časový krok t a druhá z vah v krocích t a $t - 1$ určuje míru pravděpodobnosti doporučeného výstupu $p(t, t - 1)$. Obě hodnoty jsou zpracovány v prvku stochastického zpracování a jejich kombinací je získán výstup fuzzy řízení $u'(t)$.

AEN se snaží na základě stavu systému předpovědět následné chování systému. S dodatečnou informací o chybě systému a výstupu ASN je další procedurou usuzováno o změně vah neuronové sítě ASN. Úspěšně nasazení, pro které byla síť navržena, bylo balancování tyčového vozíku. [7]

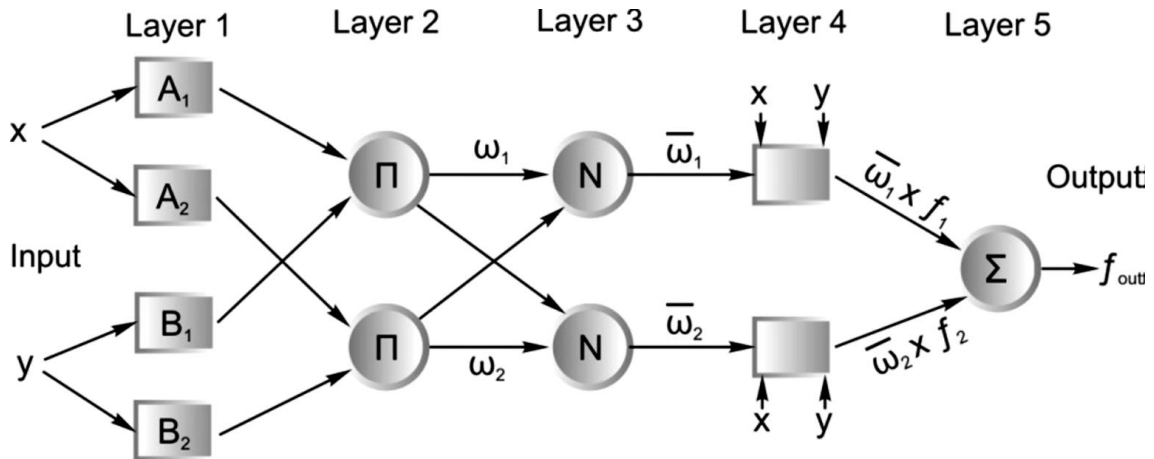


Obr. 11 Schéma Berenjiho inteligentní řízení na základě přibližného usuzování.
[7]

V předchozím textu bylo ukázáno základní dříve uvažované dělení Fuzzy-Neuro systémů. Od těchto modelů byly odvozeny další algoritmy.

4.1 ANFIS

ANFIS (Adaptivní Neuro-Fuzzy Inferenční Systém) má analogií s FIS (Fuzzy Inferenční Systém). Zásadním zlepšením v ANFIS je zapojení učení do životního procesu systému, což způsobují prvky dopředné neuronové sítě se zpětným šířením chyby do algoritmu systému. Systém se tak učí na předložená data a po dosažení požadované přesnosti lze využít jako model dat samotných. U ANFIS je využito jako základu fuzzy inference typu Sugeno. [17]



Obr. 12 Základní schéma ANFIS [17]

Schéma ANFIS (viz Obr. 12) lze popsat jako 5 vrstvý graf. Hrany tvoří spojení mezi vrstvami a uzly proces. Uzly se dělí na adaptovatelné, mění své parametry v průběhu učení, a stálé, které zůstávají učení neovlivněny a plní vždy stejnou funkci. Stálé uzly v grafu reprezentují kruhy a proměnné uzly jsou značeny obdélníky.

Během učení ANFIS dochází opakovaně ke dvěma fázím:

1. Dopředný běh – ze vstupních proměnných je prostřednictvím antecedentů vyvozen konsekvant a následným fuzzy usuzováním vystupuje ze systému ostrá hodnota výstupu.
2. Zpětný průběh – z odchylky žádaného a skutečného výstupu se zpětně šíří chyba a pomocí gradientní metody pro minimalizaci chybové funkce se upravují parametry uzlů ANFIS (obdélníky ve schématu).

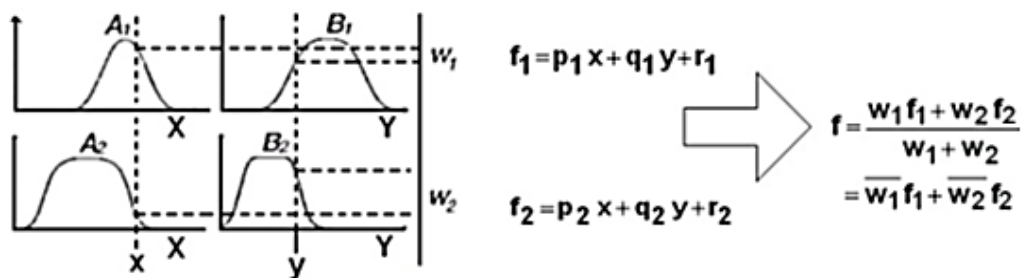
Základní popis učícího procesu

Předpokládejme jednoduchou bázi pravidel se dvěma pravidly ve tvaru Sugeno:

$$\text{IF } (x \text{ je } A_1) \text{ AND } (y \text{ je } B_1) \text{ THEN } f_1 = p_1 \cdot x + q_1 \cdot y + r_1, \quad (20)$$

$$\text{IF } (x \text{ je } A_2) \text{ AND } (y \text{ je } B_2) \text{ THEN } f_2 = p_2 \cdot x + q_2 \cdot y + r_2, \quad (21)$$

Tato pravidla jsou zobrazena na Obr. 13.



Obr. 13 Náčrt fuzzy usuzování ANFIS se dvěma pravidly [14]

Při dopředném běhu prochází data od vstupu po výstup vzestupně vrstvami 1 – 5. (viz Obr. 12). Popis vrstev včetně rovnic je čerpán ze zdroje [17].

Vrstva 1

Protože x, y nabývají ostrých hodnot, jsou fuzzyfikovány funkcemi příslušnosti ve tvaru Bellovy funkce. Bellovu funkci zapíšeme:

$$\mu_{A_i}(x) = \frac{1}{1 + \left(\frac{x - c_i}{a_i}\right)^{2b_i}}, \quad (22)$$

Případně je možné v okrajových oblastech využít:

$$\mu_{A_i}(x) = \exp\left(-\frac{x - c_i}{a_i}\right), \quad (23)$$

V této vrstvě jsou v průběhu učení adaptovány parametry a_i , b_i a c_i všech fuzzy množin. Počáteční hodnoty jsou zadány během inicializace. Musí být dodržen předpoklad: $0 \leq \mu_{A_i}(x) \leq 1$. Výstup je roven:

$$\begin{aligned} Q_{1,i} &= \mu_{A_i}(x) & \text{pro } i &= 1,2 \\ Q_{1,i} &= \mu_{B_{i-2}}(y) & \text{pro } i &= 3,4 \end{aligned} \quad (24)$$

Vrstva 2

Vrstva nemění parametry během učení. S výstupy předchozí vrstvy je proveden průnik dle (5), čímž jsou získány tzv. váhy pravidel.

$$Q_{2,i} = \omega_i = \mu_{A_i}(x) \cdot \mu_{B_i}(y) \quad \text{pro } i = 1,2 \quad (25)$$

Vrstva 3

Vrstva nemění parametry během učení. Váhy pravidel jsou normalizovány dle vztahu:

$$Q_{3,i} = \bar{\omega}_i = \frac{\omega_i}{\sum_i \omega_i} \quad \text{pro } i = 1,2. \quad (26)$$

Vrstva 4

Všechny uzly v této vrstvě jsou adaptovány při učení. Parametry p_i, q_i a r_i jsou pro každý konsekvant z pravidel (20) a (21) proměnné. Výstupem 4. vrstvy je

$$Q_{4,i} = \overline{\omega}_i f_i = \overline{\omega}_i (p_i x + q_i y + r_i) \quad (27)$$

pro $i = 1, 2$

Vrstva 5

Výstupní vrstva ANFIS je stálá a provádí součet parciálních výsledků předchozí vrstvy v celkový výstup, vyjádřeno rovnicí:

$$Q_{5,i} = f_{out} = \sum_i \overline{\omega}_i f_i \quad \text{pro } i = 1, 2 \quad (28)$$

Zpětné šíření chyby

Chyba je spočtena součtem rozdílů žádaného a skutečného výstupu ANFIS umocněných na druhou přes všechny uzly 5. vrstvy (v našem případě jednoho uzlu). Následně se chyba pomocí inverzních operací distribuuje od vrstvy 5. až k 1. Algoritmus zpětného šíření chyby je popsán ve zdroji [14], kde je dále provedeno praktické nasazení algoritmu ANFIS na rozpoznávání řeči. Další náhled na praktické výsledky ve finanční oblasti pro predikci burzovního trhu odhaluje zdroj [17][14]. Při popisu růstu kvasinek v čase na základě vhodných veličin s přímým vlivem byl úspěšně aplikován algoritmus ANFIS ve článku [20].

4.2 Fuzzy-neuronové sítě s učením bez učitele

Neuronové sítě s učením bez učitele patří do skupiny neuronových sítí, které při svém učení nemají k dispozici žádaný výstup. Takové sítě tedy v průběhu učení nepřibližují svůj výstup výstupu v datech, ale zachycují trendy, které jsou v datech nepřímo vyjádřeny.

Tyto neuronové sítě pracují na bázi shlukové analýzy. Hustota neuronové sítě se zvyšuje v oblasti shluku v datech. V případě řídké sítě se právě ve významném shluku po naučení nachází nějaký neuron, reprezentant.

Při práci s neuronovou sítí se setkáváme s pojmem životního cyklu neuronové sítě. Neuronová síť při vzniku prochází *organizační fází*. V organizační fázi je vytvořena topologie neuronové sítě a zvolen typ neuronu.

Následně inicializovaná síť přechází do *adaptační fáze*. Při adaptaci dochází k učení neuronové sítě formou úpravy vah v neuronové struktuře. U pokročilejších algoritmů dochází při adaptaci k úpravě topologie sítě za účelem přesnějšího popisu vstupních dat. Jak bylo dříve uvedeno, sítě s učením bez učitele nemají žádaný výstup v datech a při jejich učení je nutné využít vhodného algoritmu k adaptaci vah. Algoritmus adaptace nemůže pracovat s pojmem šíření chyby žádaného a skutečného výstupu, ale chybu vyjadřuje jako vzdálenost vah a předložených vzorů z tréninkových dat. Adaptace stojí na principu vektorové kvantizace. [18]

Poslední fází životního cyklu neuronové sítě je *aktivace*. V této fázi již nedochází k úpravě topologie nebo konfigurace vah. Ve fázi aktivace jsou neuronové sítě předložena data a neuronová síť určí neuron, který daná data reprezentuje.

Příkladem neuronové sítě využívané k adaptaci fuzzy systémů je Kohonenova síť.

4.3 Kohonenova síť

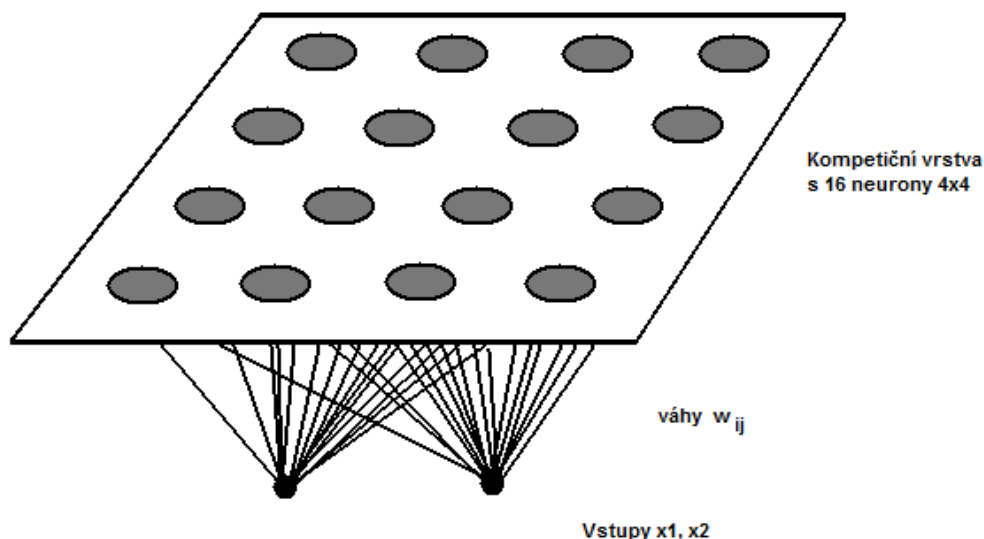
Kohonenova síť je neuronová síť s učením bez učitele založená na tzv. kompetici, soutěžení mezi neurony.

4.3.1 Životní cyklus sítě

Topologie

Kohonenova síť je složena ze dvou vrstev neuronů. V první, vstupní, vrstvě neurony mají pouze jeden vstup. Mají práh roven nule a přenosová funkce neuronů je lineární. Počet neuronů vstupní vrstvy je roven dimenzi vstupních dat, počtu atributů předkládaných dat. Druhá vrstva je tzv. Kohonenova mapa. V druhé vrstvě mají neurony stejný počet vstupů, jako je počet neuronů vstupní vrstvy. Spojení mezi vrstvami sítě je jednosměrné a informace se šíří z vstupní vrstvy do Kohonenovy mapy a každý spoj má

svoji váhu w_{ij} . Tyto váhy slouží jako paměť sítě a reprezentují polohu daného neuronu v prostoru vstupních dat.



Obr. 14 Topologie Kohonenovy sítě (zdroj [6])

V Kohonenově mapě je aktivní vždy jen jeden neuron. Jedná se o aktuálního vítěze kompetice pro předložená data. Vítězný neuron adaptuje své váhy.

Učení

Kohonenova síť se adaptuje prostřednictvím zjednodušeného Lloydova algoritmu, tzv. Kohonenova učení. Kohonenova síť při předložení vzoru spočte euklidovské vzdálenosti všech neuronů od tohoto vzoru a nalezne vítěze, nejbližšího reprezentanta, dle rovnice (29):

$$c = \arg \min_{i=1..n} \|x^{(s)} - w_i\|, \quad (29)$$

kde c je index vítěze kompetice v Kohonenově mapě s n neuronů,

$x^{(s)}$ je předložený s . vzor,

w_i jsou váhy neuronu s indexem i .

Tento vítězný neuron je pozitivně adaptován vůči předloženému vzoru dle formule:

$$w_i^{(s+1)} = w_i^{(s)} + \mu(x^{(s)} - w_i^{(s)}), \quad (30)$$

kde μ je parametr učení, $\mu \in (0,1)$.

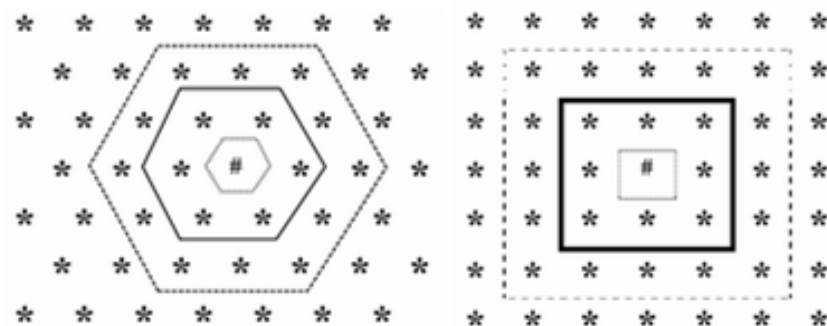
Rovnice (30) vyjadřuje, jak spočíst váhy w_i před předložením $(s+1)$. vzoru tréninkové množiny. Parametr učení není konstantní, ale mění se v průběhu iterativního procesu učení. V každé iteraci se postupně předkládají všechny vzory tréninkové množiny a na konci iterace dochází ke změně, zmenšení, hodnoty parametru učení μ a okolí vítězného neuronu ρ , přičemž musí platit:

$$\mu^{(t+1)} < \mu^{(t)} \text{ a } \lim_{t \rightarrow \infty} \mu^{(t)} = 0, \quad (31)$$

$$\rho^{(t+1)} \leq \rho^{(t)} \text{ a } \lim_{t \rightarrow \infty} \rho^{(t)} = 0, \quad (32)$$

kde t je index iterace.

Parametr ρ , okolí vítězného neuronu, je využíván při adaptaci nejen vítěze, ale také nejbližšího počtu neuronů v topologii, viz Obr. 15. Neurony v okolí vítěze se mohou adaptovat stejnou měrou jako vítěz nebo podle funkce okolí. Okolí a jeho průběh v čase je důležitým parametrem. Okolí ovlivňuje správné rozvinutí sítě a zajišťuje promítnutí sousedského vztahu z topologie sítě do sousedské polohy v prostoru vstupních dat. Zároveň volba parametrů okolí může negativně ovlivnit výslednou naučenost sítě.



Obr. 15 Okolí vítězného neuronu (#) v Kohonenově mapě
vlevo šestiúhelníková topologie, vpravo čtyřúhelníková topologie
(zdroj [6])

Krokový postup učení

1. Inicializace sítě, dle nastavení topologie. Přiřazení počátečních hodnot parametru učení a okolí. Náhodné nastavení vah v prostoru vstupních dat.
2. Předložení náhodného vzoru z tréninkové množiny.
3. Určení vítězného neuronu dle vztahu (29) a jeho adaptace rovnicí (30).
4. Opakování kroků 2. a 3. do vyčerpání všech tréninkových vzorů.
5. Úprava parametrů učení μ a okolí ρ ve smyslu rovnic (31) a (32).
6. Opakování kroků 2 – 5. dokud není splněna některá z podmínek:
 - Počet iterací překročí stanovenou hranici $(t) > (t)_{max}$.
 - Hodnota parametru učení je dostatečně malá $\mu^{(t+1)} < \mu_{min}$.
 - Chyba pokrytí tréninkových dat $E = \frac{1}{p} \sum_{s=1}^p \|x^{(s)} - w_c\|^2$ je menší než stanovená dosažitelná hodnota ε . $E < \varepsilon$.

Aktivace

Algoritmus aktivace Kohonenovy sítě je ve srovnání s adaptací prostší. Při předložení vstupních dat je proveden výpočet a určení vítězného neuronu dle vztahu (29). Nyní již nepracujeme s pojmem okolí, ale pouze vítěz se přihlásí jako reprezentant daných dat.

4.3.2 Doporučené nastavení Kohonenovy sítě

Ve své předchozí práci jsem se zabýval analýzou vlivu počátečního nastavení vah na výslednou kvalitu sítě, viz zdroj [6]. Tato práce vyústila v citlivostní analýzu a následné stanovení vhodných parametrů Kohonenovy sítě za účelem minimalizování vlivu počátečního nastavení vah na výsledný stav sítě. Tyto parametry měly umožnit dosažení nejvyšší možné přesnosti reprezentace dat a zároveň omezení nežádoucích jevů jako částečné a úplné překroucení sítě.

Funkce parametru učení byla ve tvaru:

$$\mu(t) = \mu_0 \cdot e^{-k_\mu t}, \quad (33)$$

kde μ_0 je počáteční hodnota parametru učení $\mu_0 \in (0,1)$,
 k_μ je koeficient exponenciály učení.

Jednodušší alternativou je funkce lineární klesající ve tvaru:

$$\mu(t) = \mu_0 - t \frac{\mu_0}{(t)_{max}} \text{ nebo } \mu(t+1) = \mu_0(t) - \varepsilon_\mu, \quad (34)$$

kde ε_μ je malá konstanta.

Funkce okolí byla definována skokovou funkcí:

$$\rho(t) = \rho_0 - k_\rho t, \quad (35)$$

kde $\rho(t)$ okolí nabývalo nejbližších vyšších nezáporných celých čísel,
 ρ_0 je počáteční hodnota okolí,
 k_ρ je krok okolí.

Z citlivostní analýzy vyplynula následující doporučení:

1. Počáteční hodnota okolí ρ_0 by měla být větší než polovina počtu neuronů na delší straně topologie v Kohonenově mapě.
2. Počáteční hodnotu parametru učení μ_0 pro vztah (33) je vhodné volit od 0,6 do 1, pro rozmanité tréninkové datové soubory se osvědčila hodnota $\mu_0 = 0,8$.
3. Nastavení parametrů k_μ a k_ρ je vzájemně silně závislé, protože jejich nastavení ovlivňuje, zdali bude mít síť dostatek hybnosti přesně aproximovat data po dosažení okolí $\rho = 0$ a zároveň ovlivňuje počet iterací potřebný k naučení sítě. Obtížnost volby je umocněna vlivem počtu tréninkových vzorů na tyto parametry. Pokud je tréninkový soubor dostatečně velký řádově, alespoň 10x větší než počet neuronů, potom lze zvolit vyšší k_μ a nižší k_ρ , což značně urychlí učicí proces. Osvědčila se konfigurace $\mu_0 = 0,8$; $k_\mu = 0,07$; $\rho_0 = 6$; $k_\rho = 4$ pro síť 8x8 neuronů a 214 tréninkových vzorů (náhodná generace).
4. Inicializovat váhy sítě je doporučeno v krajní oblasti prostoru vstupních dat a na řádově menší hodnoty než je rozsah dat, např. data 0-1, váhy 0-0.01 pro všechny atributy.
5. Nelze však opomenout zásadní neduh Kohonenovy neuronové sítě, kterým je citlivost na pořadí předkládaných vzorů. Tento nedostatek je vlastností algoritmu daný adaptací po každém tréninkovém vzoru. Nepodařilo se potlačit vliv pořadí předložených vzorů vhodným nastavením parametrů sítě.

4.3.3 Aplikace Kohonenovy sítě

Jak plyne z algoritmu a výsledku adaptace, Kohonenova síť je určena ke shlukové analýze rozsáhlých datových souborů. Výsledkem procesu učení je soubor reprezentantů původních dat, který je mnohem menší než původní datový soubor. Dle nastavení topologie máme n vektorů o stejné délce jako vektor každého z vstupních dat. Této vlastnosti se využívá při kompresi obrázků, barevného spektra nebo obecně dat.

Aktivační fáze ovšem nabízí jiné vlastnosti sítě. Z báze dat ve formě naučených vah sítě je po předložení vstupního vzoru nalezen vítěz, reprezentant, který může představovat anomálii v řeči jedince a odhalit tak nemocné dítě, viz [13]. Dále je využíváno Kohonenovy sítě k analýze chování zákazníků a jejich následné klasifikaci. Nelze opomenout úspěšné nasazení při odhalování objektů v obrazech a videích, či překladu psaného textu.

4.3.5 Modifikace základního algoritmu

Na základě aplikací se v literatuře objevuje několik možných úprav Kohonenovy sítě, resp. jejího učení, za účelem zpřesňování výsledků.

Metoda LVQ

Pomocí této modifikace je Kohonenova síť upravena tak, že se doučuje a zpřesňuje svůj výstup vůči požadovanému výstupu v datech. To znamená, že data pro tento algoritmus obsahují také žádané výstupní hodnoty na rozdíl od původního Kohonenova učení. Metoda LVQ byla čerpána ze zdroje [18]. Cyklus učení s metodou LVQ je následující:

1. Učení bez učitele – Kohonenovo učení dle kapitoly 4.3.1.
2. Přiřazení tříd/výstupních hodnot jednotlivým reprezentantům
3. Doučení metodou LVQ

V bodě 2 provádíme přiřazování tříd na základě četnosti vítězství jednotlivých neuronů pro danou třídu. Každému neuronu se sečtou četnosti vítězství u tréninkových dat každé z tříd a neuron se stává reprezentantem nejčetněji zastoupené třídy.

Následné doučování probíhá podle 3 možných schémat:

LVQ 1 – jsou předkládány tréninkové vzory a u vítězného neuronu dochází ke kontrole, zda reprezentuje třídu, která je obsažena ve vzoru. Pokud si třídy odpovídají, jsou váhy pozitivně adaptovány směrem ke vzoru. V opačném případě jsou váhy inhibovány, tedy neuron se odsouvá ve směru přímo od vzoru. Viz rovnice:

$$w_{ij}^{(s+1)} = \begin{cases} w_{ij}^{(s)} + \mu_{LVQ} (x_j^{(s)} - w_{ij}^{(s)}), & d^{(s)} = y^{(s)} \\ w_{ij}^{(s)} - \mu_{LVQ} (x_j^{(s)} - w_{ij}^{(s)}), & d^{(s)} \neq y^{(s)} \end{cases}, \quad (36)$$

kde μ_{LVQ} je parametr doučení metodou LVQ,
 $d^{(s)}$ je žádaný výstup (data),
 $y^{(s)}$ je skutečný výstup (sítě).

LVQ 2 – rozšiřuje variantu LVQ 1 o vliv na nejbližší neuron opačné třídy než je aktivní vítěz. Opět pro oba neurony platí adaptace dle rovnice (36). Nevýhodou algoritmu je, že nezáleží, který z neuronů je skutečně nejbližší a může dojít k odstranění neuronu mimo data a znehodnocení výsledného stavu sítě.

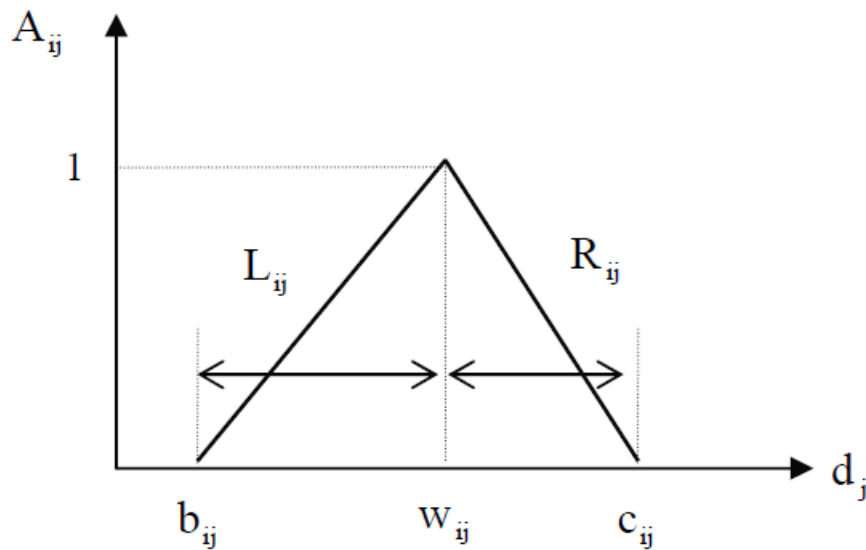
LVQ 3 – odstraňuje negativa modifikace LVQ 2 a posiluje pozici všech reprezentantů třídy shodné se vzorem.

4.3.6 Tvorba neuro-fuzzy klasifikátoru na Kohonenově síti

Výstupem učení Kohonenovy sítě je matice vah. Tyto váhy představují polohy clusterů. Váhy neuronů využijeme jako středy fuzzy bázových buněk. Často využívaná metoda stavby fuzzy množiny na základě Kohonenovy sítě je popsán v článku [1]. Článek je zaměřen na mapování a tvorbu map mobilního robota na základě ultrazvukových snímačů vzdálenosti a následné rozpoznávání situací. Analogickému algoritmu se věnoval Ing. Ondřej Machek ve své diplomové práci na FEKT VUT v Brně pod vedením prof. Ing. Pavla Jury, CSc. [13].

Popis Algoritmu

Algoritmus je z části analogický Kohonenově síti, ale během iterací a předkládání vzorů se adaptují navíc parametry L_{ij} a R_{ij} . Oba parametry slouží jako pomocné hodnoty při stavbě trojúhelníkových funkcí příslušnosti.



Obr. 16 Trojúhelníková funkce příslušnosti v ose d_j (x_j) s vrcholem v w_{ij} a rameny: levým $L_{ij} = w_{ij} - b_{ij}$ a pravým $R_{ij} = c_{ij} - w_{ij}$ [1]

Postup je v následujícím pořadí:

1. Inicializace sítě, dle nastavení topologie. Přiřazení počátečních hodnot parametru učení a okolí. Náhodné nastavení vah v prostoru vstupních dat. Inicializace matic L a R na malé hodnoty.
2. Předložení náhodného vzoru x z tréninkové množiny.
3. Určení vítězného neuronu dle vztahu (29) a jeho adaptace rovnicí (30).
4. Výpočet prostých vzdáleností vah a vzoru Δ_{ij} pro všechny neurony:

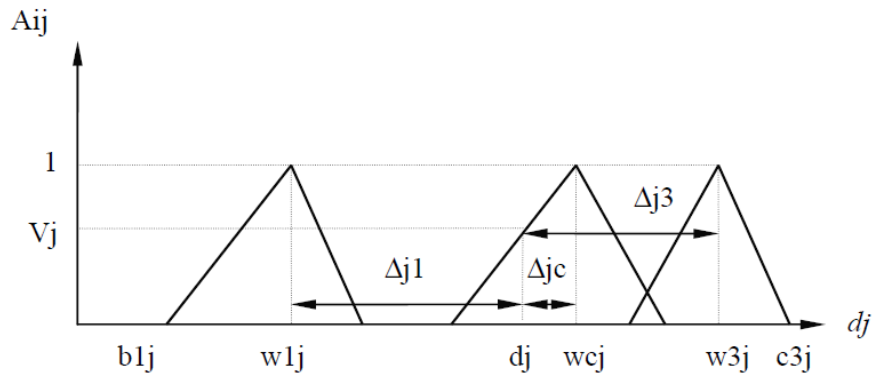
$$\Delta_{ij} = |x_j - w_{ij}| \quad (37)$$

5. Určení sumy všech odchylek:

$$S_j = \sum_i^n \Delta_{ij} \quad (38)$$

6. Stanovení hodnoty funkce příslušnosti V_j vítězného neuronu (Δ_{cj}) pro j . složku vstupních dat (viz Obr. 17):

$$V_j = 1 - \frac{\Delta_{cj}}{S_j} \quad (39)$$



Obr. 17 Ilustrace výpočtu hodnoty příslušnosti k množině A_{cj} [1]

7. Aktualizace složek vítěze v matici L :

$$\begin{aligned} L_{cj}^{(s+1)} &= L_{cj}^{(s)} + \mu \left(\frac{w_{cj}^{(s)} - x_j^{(s)}}{1 - V_j^{(s)}} - L_{cj}^{(s)} \right) \quad \text{pro } x_j < w_{ij}, \\ L_{cj}^{(s+1)} &= L_{cj}^{(s)} \quad \text{pro } x_j \geq w_{ij}, \end{aligned} \quad (40)$$

A matici R :

$$\begin{aligned} R_{cj}^{(s+1)} &= R_{cj}^{(s)} + \mu \left(\frac{w_{cj}^{(s)} - x_j^{(s)}}{1 - V_j^{(s)}} - R_{cj}^{(s)} \right) \quad \text{pro } x_j > w_{ij}, \\ R_{cj}^{(s+1)} &= R_{cj}^{(s)} \quad \text{pro } x_j \leq w_{ij}, \end{aligned} \quad (41)$$

8. Opakování kroků 2. až 7. do vyčerpání všech tréninkových vzorů.

9. Úprava parametrů učení μ a okolí ρ ve smyslu rovnic (31) a (32).

10. Opakování kroků 2 – 5. dokud není splněna některá z podmínek:

- Počet iterací překročí stanovenou hranici $(t) > (t)_{max}$.
- Hodnota parametru učení je dostatečně malá $\mu^{(t+1)} < \mu_{min}$.
- Chyba pokrytí tréninkových dat $E = \frac{1}{p} \sum_{s=1}^p \|x^{(s)} - w_c\|^2$ je menší než stanovená dosažitelná hodnota ε . $E < \varepsilon$.

4.4 Fuzzy Kohonen Clustering Network

Metoda FKCN (Fuzzy Kohonen Clustering Network) vychází z teorie metody fuzzy c-means (FCM) ve spojení s Kohonenovou clusterovací sítí (KCN). Metoda FKCN využívá fuzzy příslušnosti z paralelního výpočtu FCM jako parametru učení neuronů Kohonenovy sítě. Algoritmus popisuje autor ve článku [2], kde dále rozvádí teorii na intervalová data.

Topologie FKCN odpovídá Kohonenově síti, ale hlavní rozdíl je v adaptační fázi sítě. Průběh učení popisují následující body:

1. Inicializace vah neuronové sítě, nastavení minimální krokové odchylky vah sítě ε_w , stanovení maximálního počtu iteračních kroků $(t)_{max}$ a počáteční hodnoty fuzzy parametru m_0 .
2. Spočtení fuzzy parametru m :

$$m^{(t)} = m_0 - t \cdot \frac{m_0 - 1}{t_{max}}, \quad (42)$$

3. Výpočet fuzzy příslušnosti všech n neuronů k předloženému vzoru (s):

$$u_i^{(s)}(t) = \left[\sum_{h=1}^n \left(\frac{\sum_{j=1}^p (x_j^{(s)} - w_{ij}(t-1))^2}{\sum_{j=1}^p (x_j^{(s)} - w_{hj}(t-1))^2} \right)^{\frac{1}{m^{(t)}-1}} \right]^{-1}, \quad (43)$$

4. Výpočet fuzzyfikovaného parametru učení θ pro všechny neurony:

$$\theta_i^{(s)}(t) = \left(u_i^{(s)}(t) \right)^{m^{(t)}}, \quad (44)$$

5. Opakování bodů 3. a 4. pro všechny tréninkové vzory r .
6. Aktualizace vah FKCN dle vztahu:

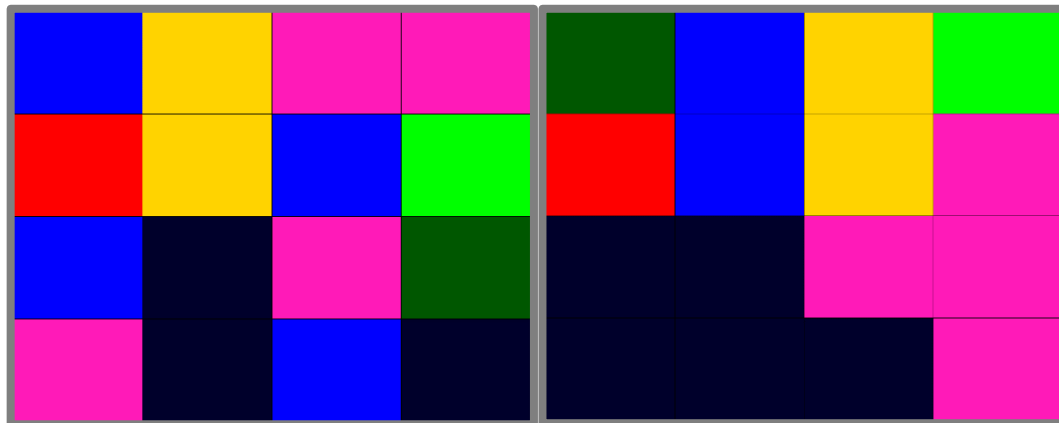
$$\mathbf{w}_i(t) = \frac{\sum_{s=1}^r \theta_i^{(s)}(t) \mathbf{x}^{(s)}}{\sum_{s=1}^r \theta_i^{(s)}(t)}, \quad (45)$$

7. Výpočet změny vah mezi kroky:

$$E_t = \|\mathbf{w}(t) - \mathbf{w}(t-1)\|^2, \quad (46)$$

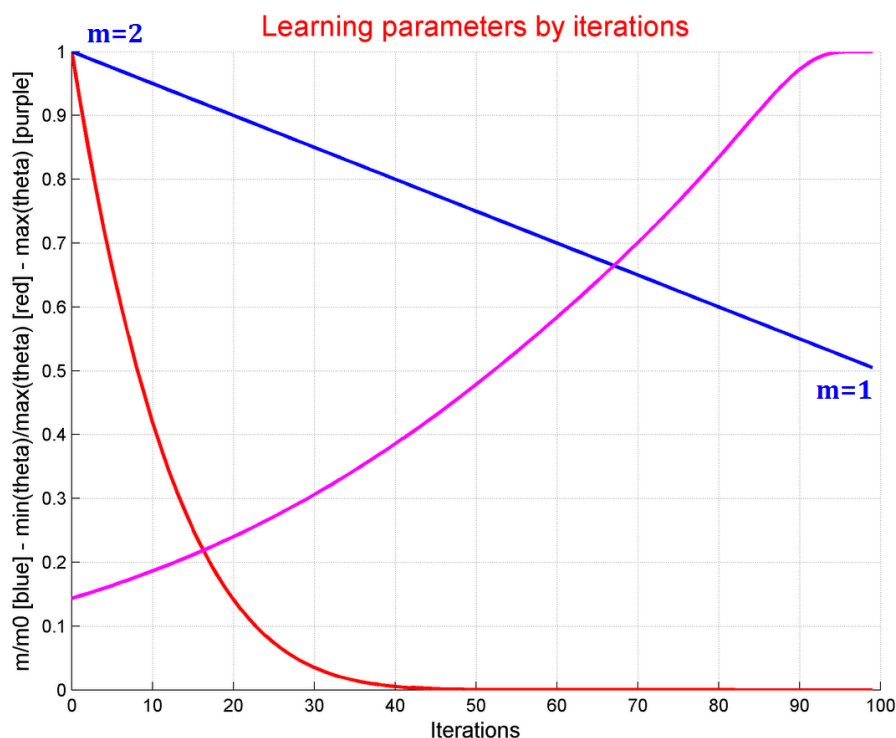
8. Pokud je $E_t > \varepsilon_w$ nebo $(t) < (t)_{max}$ opakuj kroky 2. – 8.

Výhodou algoritmu je, že minimalizuje vliv pořadí předložení tréninkových vzorů a zvyšuje rychlost konvergence. I když v názvu algoritmu vystupuje Kohonenova síť, algoritmus nepracuje s okolím vítězného neuronu ve smyslu vazeb mezi neurony, ale okolí v prostoru vah. Z těchto vlastností algoritmu plyne, že následné zobrazení Kohonenovy mapy formou barevně zbarvených polí, coby struktury vazeb neuronové sítě, nemá smysl a je zavádějící, viz Obr. 18.



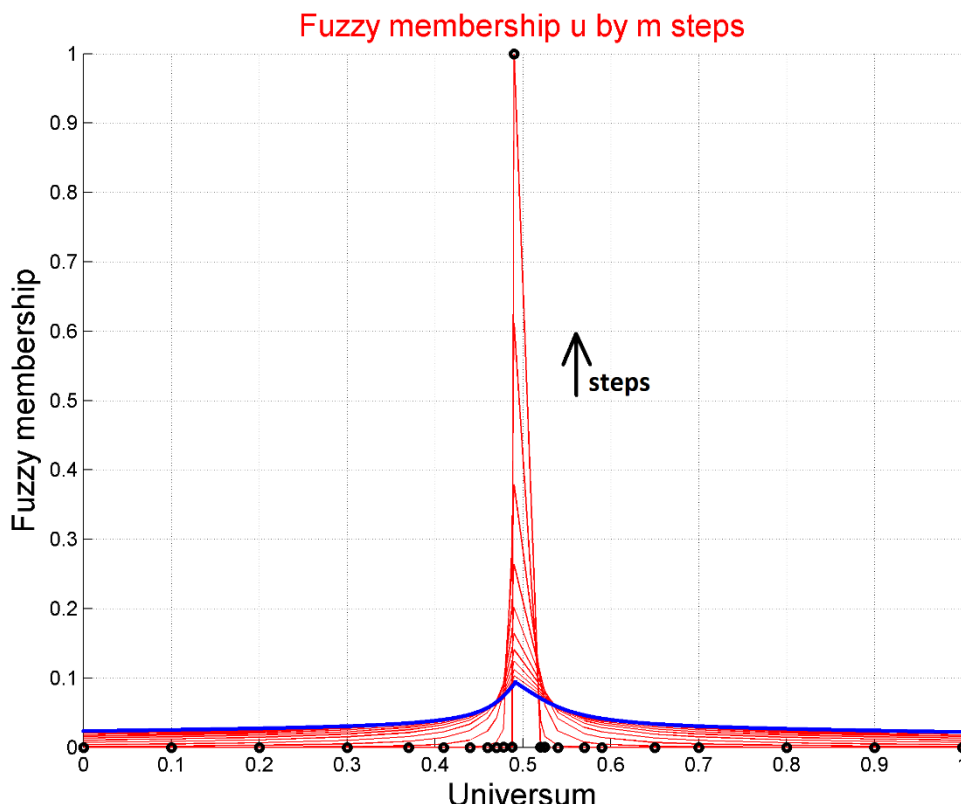
Obr. 18 Srovnání FKCN bez vlivu vazeb v neuronové struktuře (vlevo) a s vlivem vazeb mezi neurony (vpravo) pro data s 300 vzory a 7 shluky různých tříd (barvy), síť 4x4 neurony.

Na Obr. 18 vlevo je patrné, že reprezentanti stejné třídy ve většině případů nemají společné vazby. Průběh učicích parametrů FKCN pro lineární klesající funkci fuzzy parametru m je na Obr. 19.



Obr. 19 Průběh učení FKCN

V grafu na Obr. 19 je modře znázorněna křivka m/m_0 a na její hodnotu reaguje křivka maximální hodnoty parametru učení θ (fialová). Červeně je vyznačen poměr minimální a maximální hodnoty parametru učení $\theta_{\min}/\theta_{\max}$. Tento graf byl vytvořen za účelem nahlédnutí do průběhu učení a jako základ sloužil soubor neuronů a jednoho vzoru (viz Obr. 20)



Obr. 20 Graf fuzzy příslušnosti vzoru ($x=0.5$) k neuronům s váhami $w_N = \{ 0, 0.1, 0.2, 0.3, 0.37, 0.41, 0.44, 0.46, 0.47, 0.478, 0.488, 0.49, 0.52, 0.525, 0.54, 0.57, 0.59, 0.65, 0.7, 0.8, 0.9, 1 \}$

V grafu (Obr. 20) je vykreslen průběh rozložení fuzzy příslušnosti po okolních neuronech. Červeně jsou vykresleny křivky fuzzy příslušnosti a neurony ve finálním stavu zvýrazňují černé kruhy. Počáteční stav pro $m_0 = 6$ je zvýrazněn modrou křivkou. Křivky se pro okolí postupně přimykají k ose universa. Pro $m \rightarrow 1$ jediným neuronem s nenulovou hodnotou příslušnosti zůstává vítězný neuron ($w = 0.49$) s hodnotou příslušnosti ($u = 1$). Pro případ, že by byla počáteční hodnota parametru m nastavena $m \rightarrow \infty$ potom má vzor fuzzy příslušnost ke všem neuronům stejnou a to $u = 1/n$, kde n je počet neuronů.

4.4.1 Modifikace FKCN

Protože pro klasifikaci s asistencí lidského personálu je vhodné, aby zobrazení dávalo lepší smysl a třídy byly reprezentovány sousedícími prvky i na obrazu, byla provedena úprava v algoritmu, která vzájemné vazby neuronů zohledňuje. Při stanovování fuzzy příslušnosti neuronů k předloženým datům jsou započítány pouze ty, které s vítězem přímo sousedí v rozsahu okolí ρ . Tato úprava je provedena v rovnici (43), kdy platí:

$$u_i^{(s)}(t) = \left[\sum_{h=1}^n \left(\frac{\sum_{j=1}^p \left(x_j^{(s)} - w_{ij}(t-1) \right)^2}{\sum_{j=1}^p \left(x_j^{(s)} - w_{hj}(t-1) \right)^2} \right)^{\frac{1}{m^{(t)}-1}} \right]^{-1} \quad \text{pro } |i - c| \leq \rho, \quad (47)$$

$$u_i^{(s)}(t) = 0 \quad \text{pro } |i - c| \geq \rho$$

kde c je index neuronu s nejvyšší hodnotou příslušnosti $u_i^{(s)}$.

Tato úprava vede opravdu ke zkvalitnění ve smyslu promítnutí vazeb sítě do datového prostoru, avšak má negativní dopad na odolnost vůči vlivu pořadí předkládání vzorů při učení sítě. Experiment byl proveden s parametrem okolí $\rho = 1$.

4.4.2 Aplikace FKCN

Fuzzy Kohonenova clusterovací síť byla poprvé představena J. C. Bezdekem v roce 1992 [4]. Od té doby nalezla celou řadu uplatnění v robotice, např. řízení kolečkového křesla pomocí náklonu invalidního uživatele na sedačce, viz [9]. Nachází uplatnění v oblasti segmentace obrazu, což dokládají zdroje [1], [6].

5 APLIKACE ZVOLENÉ METODY

Pro praktické ověření na datech jsem zvolil metodu Fuzzy-Neuro klasifikátoru na bázi Kohonenovy sítě popsanou v kapitole 4.3.6. Tento klasifikátor je prvotně navržen jako překladač pravidel obsažených v datech formou shluků na soubor fuzzy množin, resp. fuzzy bazových funkcí (FBF).

Výhodou představeného algoritmu je promítnutí lokální hustoty neuronové sítě také do výstupních pravidel a fuzzy množin formou středu a šířky základny fuzzy množin. V hustější oblasti dat jsou základny fuzzy množin užší.

Programové prostředí zvolené pro vytvoření testovacího skriptu bylo MATLAB 7.12.0. Formou funkcí a hlavního skriptu byly vytvořeny algoritmy Kohonenovy sítě s algoritmem LVQ, FKCN s následným doučením s výběrem prvků patřící k třídě a jako poslední zmíněný Fuzzy-Neuro klasifikátor dle článku [1].

5.1 Cíle praktického nasazení vybraného algoritmu

Primárním cílem této práce bylo vytvořit a ověřit algoritmus, který bude vykazovat dobré klasifikační schopnosti pro náhodně generovaná data s daty obsahujícími významné shluky. Tento algoritmus musí mít schopnost učení na předložená data a následně bude možné jej vyjádřit formou fuzzy pravidel. Tato fuzzy pravidla je možné vnořit do fuzzy systému, který efektivněji provádí dopředné operace při klasifikačním procesu.

5.2 Tvorba datových souborů

Algoritmus vytvořený v MATLABU předpokládá znalost rozsahu dat a pracuje s daty na rozsahu 0-1 ve všech dimenzích. Teoreticky je počet dimenzí neomezený, ale pro snazší zobrazení byla v této práci generována data do tří dimenzionálního prostoru. 4. dimenzi tvoří výstupní hodnota, třída dat.

Pro klasifikaci vybraným algoritmem jsem nevyužil žádná dostupná lékařská, vědecká či jiná data, ale vytvořil jsem vlastní algoritmus. Algoritmus se nachází v příloze na CD pod názvem *Data_Script_141212.m*.

Vstupy algoritmu

<code>data_dim</code>	Počet dimenzí D v datech
<code>pocet_vzoru</code>	Počet požadovaných vzorů datového souboru
<code>pocet_trid</code>	Počet tříd generovaných do datového souboru
<code>vahova_malice</code>	Řádkový vektor o délce totožné s počtem tříd. i . prvek reprezentuje poměr $(0,1)$ v jakém bude i . třída k celkovému počtu datových vzorů.

`osova_rozlisitelnost` Vektor s D prvky. Vzájemný poměr prvků udává eliptičnost generovaného shluku (odchylka v rozptylu).

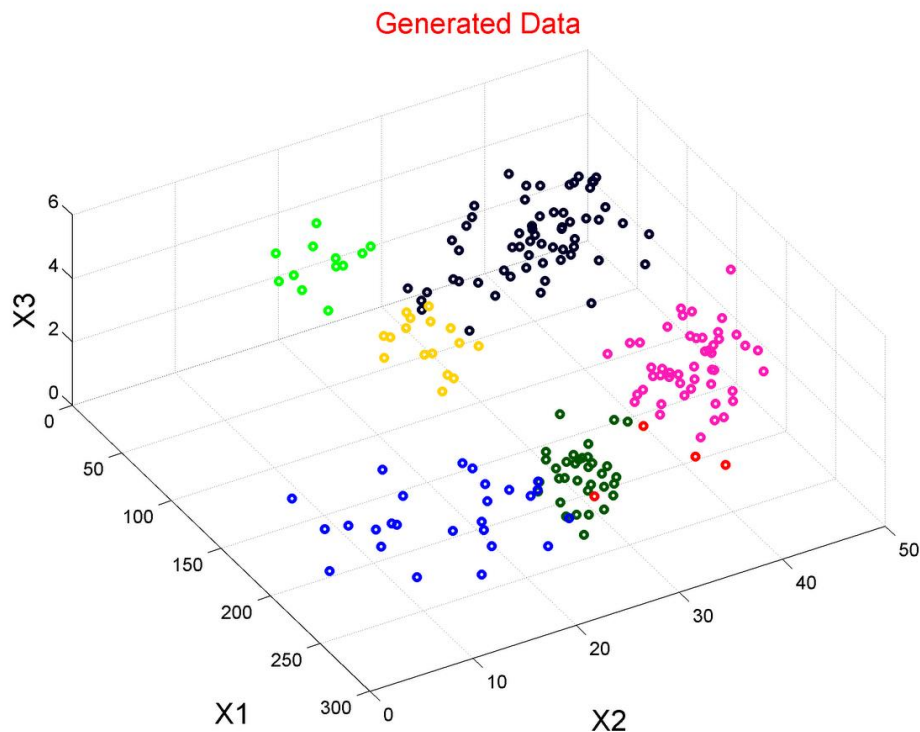
`rozmary` Vektor s D řádky a dvěma sloupci. Pro každý řádek obsahuje [minimální hodnotu, maximální hodnotu dat]

`koef_odlisitelnosti_shluku` Koeficient zapracovaný do algoritmu, který přímo ovlivňuje minimální vzdálenost mezi středy generovaných shluků.

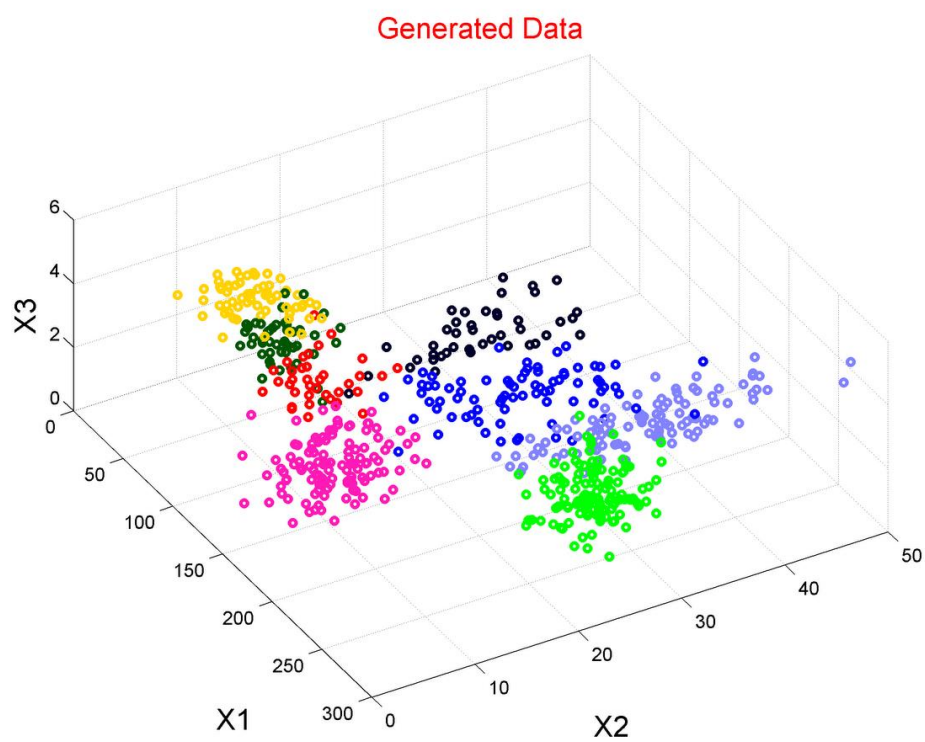
Ze zadaných parametrů je generován soubor středů datových shluků a soubor rozptylů. MATLAB nabízí funkci na generování n -rozměrných dat ze středů, rozptylů a počtu vzorů:

```
dmvnrn(mu(o,:),sig,floor(pocet_vzoru*vahova_matice(o)));
```

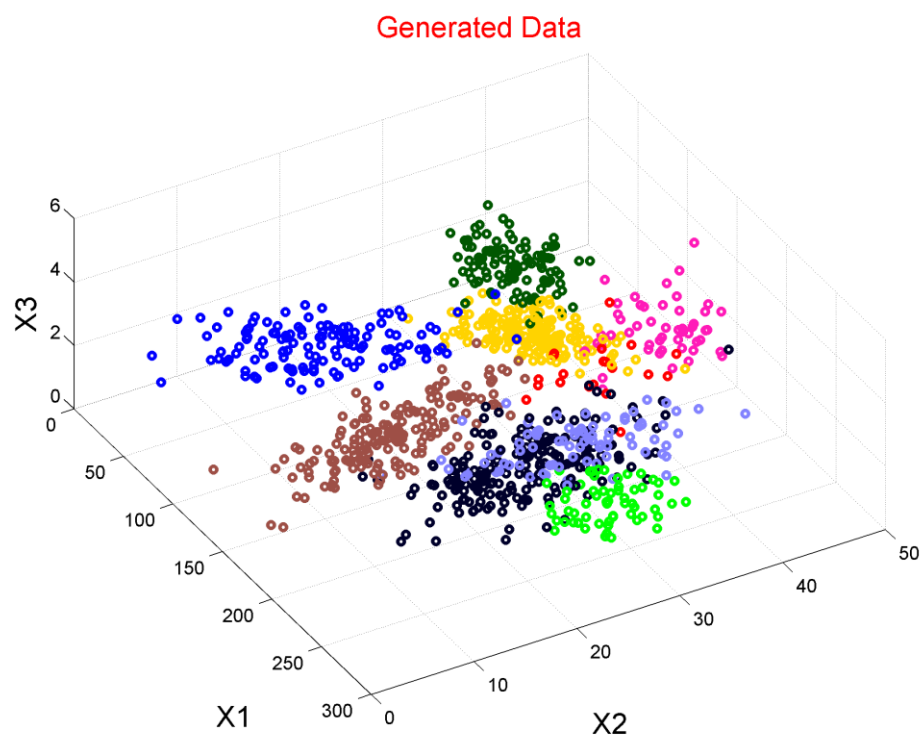
Před uložením dat do souboru `data.mat` proběhne finální kontrola a oprava rozsahu dat kontrakcí. Výstup algoritmu je možné zviditelnit proměnnou `viditelne` na `true`. Data se objeví v grafu, jak ukazuje Obr. 21, Obr. 22 a Obr. 23.



Obr. 21 Generovaná data – 200 vzorů v 7 třídách (barevně odlišeno) – dobře odlišitelné shluky.



Obr. 22 Generovaná data – 600 vzorů v 8 třídách (barevně odlišeno)
– Středně odlišitelné shluky.



Obr. 23 Generovaná data – 1000 vzorů v 9 třídách (barevně odlišeno)
– Středně odlišitelné shluky.

5.3 Učení algoritmu

Proměnné učení, které využíváme cyklicky, jsou inicializovány funkcí `initialize_fuzzynet`. Učení bez učitele zvoleného algoritmu bylo již popsáno kapitolou 4.3.6. Během učení se zároveň modifikují základny fuzzy množiny, jejichž středy jsou vyjádřeny vahami sítě. Tento proces v programu zastupuje funkce `learn_fuzzynet`. Výsledek ovšem nic nevypovídá o klasifikační schopnosti, pokud nezapojíme žádaný výstup v datech. Proto je v programu vytvořena funkce zařazení třídy k neuronu:

`classify_fuzzynet`

Funkce sečte počet vzorů, jednotlivých tříd, které jsou zastoupeny daným reprezentantem. Pouhý počet zastoupených vzorů však nezohledňuje, jakou měrou je ve skutečnosti daná třída reprezentována právě vybraným neuronem, protože třídy nejsou v datech reprezentovány se stejnou četností. Funkce `classify_fuzzynet` tento fakt zohledňuje a dělí jednotlivé četnosti celkovým počtem vzorů dané třídy. Třída nejsilněji reprezentovaná tímto neuronem je klasifikována jako výstupní třída daného neuronu, reprezentanta. V případě, že některá třída nemá žádného reprezentanta, hledá se z již zařazených neuronů ten, který tuto třídu nejlépe popisuje. Pokud existují dvě třídy takové, že existuje právě jeden reprezentant, který je v obou třídách nejsilnější a právě jediný možný reprezentující, potom může dojít k zacyklení. Tomu se algoritmus brání a po několika cyklech zanechá jednu z tříd nepřirazenou. Bohužel tato pojistka zvyšuje chybu klasifikace, ale má pozitivní vliv na robustnost algoritmu.

`score_fuzzynet`

Již ve funkci `classify_fuzzynet` je vítěz vyhodnocen na základě míry příslušnosti předloženého vzoru k jednotlivým fuzzy množinám. Míra příslušnosti se spočte pomocí násobení fuzzy příslušností jednotlivých dimenzí:

$$\mu_{A1} = \mu_{A1}^1 \cdot \mu_{A1}^2 \cdot \dots \cdot \mu_{A1}^p. \quad (48)$$

Následně je vytvořena matice pro všechny fuzzy bazové funkce, reprezentanty, A_1, A_2, \dots, A_n . Třída reprezentovaná FBF s nejvyšší hodnotou příslušnosti se porovná s třídou předloženého vzoru a správně klasifikovaný vzor inkrementuje proměnnou pozitivních výsledků. Výsledná hodnota je podělena počtem vzorů a výstup funkce `score_fuzzynet` nabývá hodnot od 0 do 1. 1 je bezchybný výsledek 100 % přesná klasifikace.

lvq_fuzzynet

Datový soubor obsahuje žádaný výstup formou třídy vzoru. V rámci experimentálního nasazení jsem k algoritmu doplnil doučovací funkci *lvq_fuzzynet*, která je motivována metodou LVQ1 učení s učitelem pro samoorganizující neuronové sítě. Tento algoritmus je popsán již v kapitole 4.3.5.

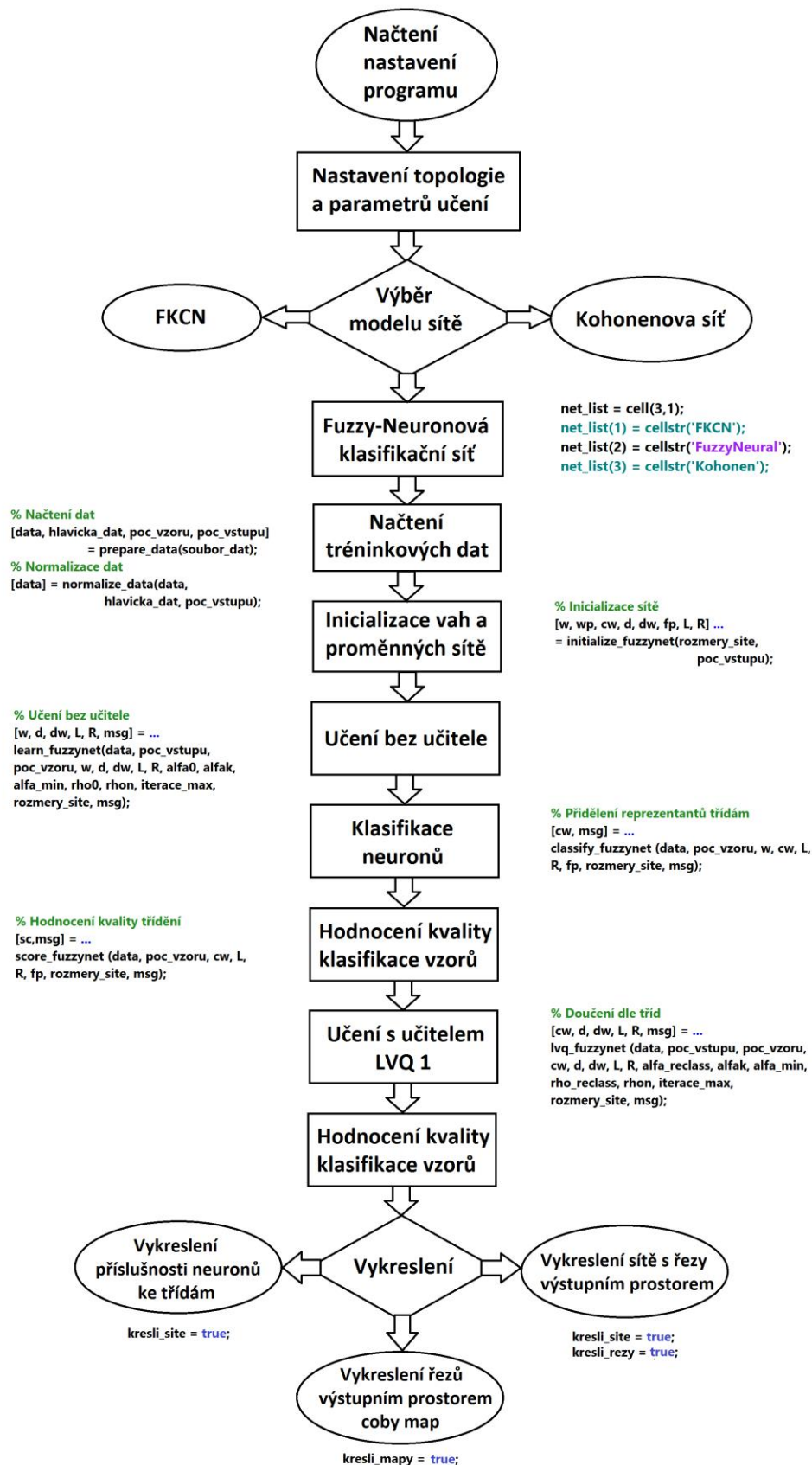
Metoda LVQ2 byla také testována. Pokus aplikovat LVQ2 na adaptaci parametrů fuzzy množin L a R byl o málo úspěšnější než aplikace na adaptaci samotných vah, ale celkově metoda LVQ2 vnášela nestabilitu do algoritmu a značně zvyšovala rozptyl kvality výsledku učení.

Metoda LVQ1 byla úspěšně nasazena jak na váhy sítě, tak na parametry fuzzy množin L a R . Tyto parametry byly na rozdíl od základního algoritmu adaptovány pouze tehdy, když třída předloženého vzoru odpovídala třídě vítěze kompetice.

Výsledek LVQ algoritmu je opět ohodnocen prostřednictvím funkce *score_fuzzynet*. Výsledná data je možné vykreslit prostřednictvím změny proměnných v úvodu programu:

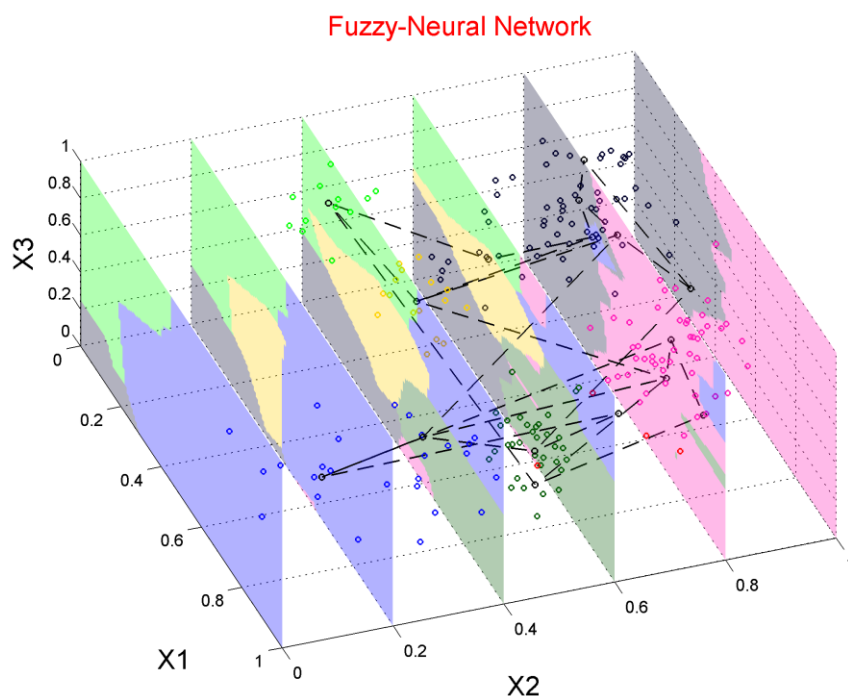
- `viditelne` – vykreslení grafů proběhne v aktivním okně pro `true`.
- `kresli_site` – vykreslí graficky neuronovou síť včetně vazeb a dat a uloží ji do podsložky programu `/pictures` pro `true`. Spolu s grafem výsledné sítě vykreslí obdélníkovou mapu reprezentantů ve stejných barvách, jako jsou vykreslena data příslušné třídy. A jako poslední vykresluje graf data + neuronová síť před učením LVQ, na kterém je možné pozorovat, jak byl stav sítě metodou LVQ ovlivněn.
- `kresli_rezy` – je-li tento parametr `true`, potom v síti výsledného stavu po LVQ učení budou zobrazeny řezy datovým prostorem se zvýrazněnou příslušností k třídám opět plochy barevně odpovídající datům příslušné třídy. Počet řezů je dán parametrem `poc_rezu`.
- `kresli_mapy` – jedná se o analogické zobrazení k řezům, ale tyto řezy budou jednotlivě zobrazeny formou subplotů. V případě řezů i map musí dojít k výpočtu hodnot funkce příslušnosti v celém prostoru. Počet těchto hodnot, tedy hustotu sítě, ovlivňuje parametr `deleni_map`.

Schéma celého algoritmu, pro fuzzy-neuronovou klasifikační síť jen na Obr. 24. Alternativní algoritmy FKCN a Kohonenova síť jsou od načtení dat identické, pouze se liší parametry funkcí a jejich zápisem (např. `learn_fuzzynet` → `learn_kohonen` nebo `learn_fkcn`). Název hlavního skriptu je *a150728_Networks_TestScript.m*.

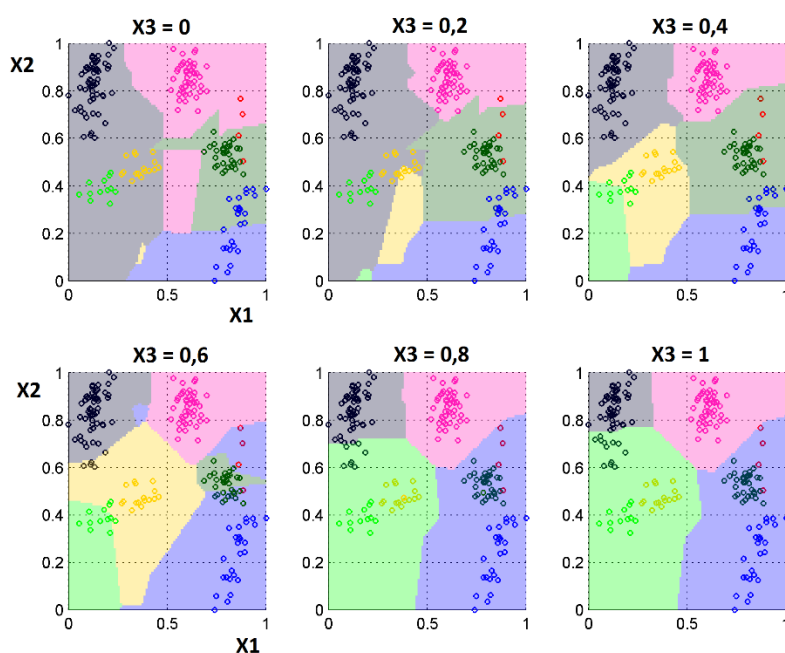


Obr. 24 Diagram běhu hlavního programu pro testování učících algoritmů

Výstupem zobrazovacích funkcí jsou následující obrázky. Na Obr. 25 je vykreslen výsledný stav sítě pro 200 datových vzorů, dobře odlišitelných 7 tříd. Výsledná kvalita klasifikace pro 4x4 neurony byla 93,5 %. Dalším grafickým výstupem jsou mapy řezů. V případě 6 byly řezy vedeny kolmo k ose X2. Pro mapy platí, že řezy jsou kolmé k ose X3 (viz.Obr. 26)

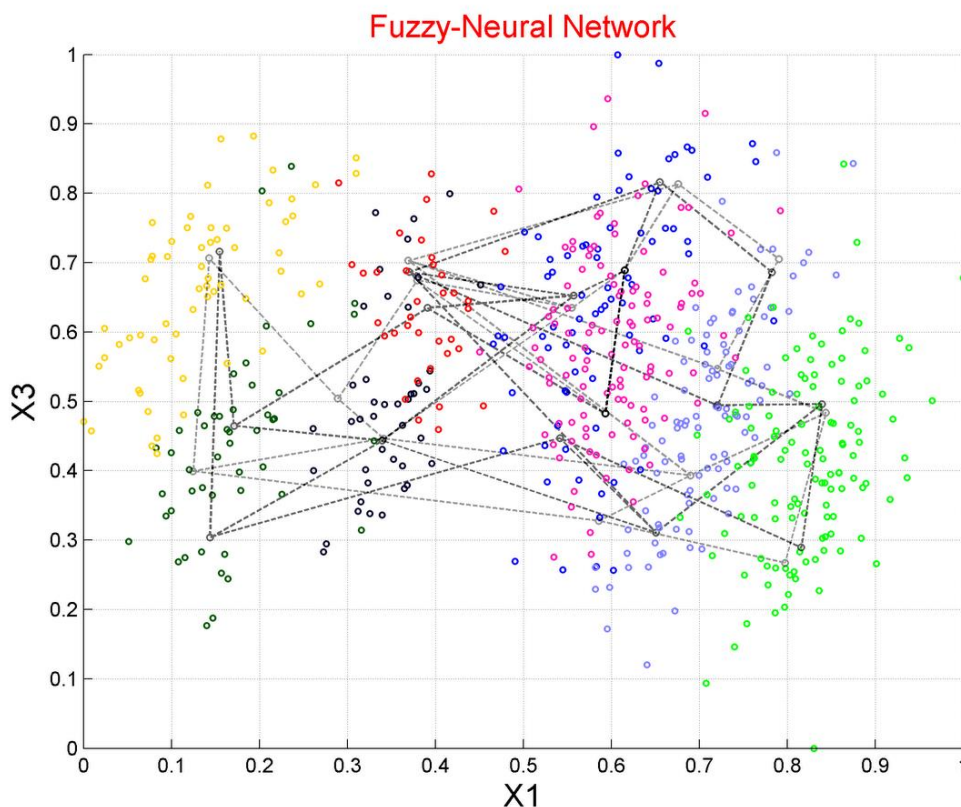


Obr. 25 Zobrazení výsledného stavu fuzzy-neuronové sítě v prostoru dat s vizualizací neuronů, vazeb v síti, tréninkových dat a řezů.



Obr. 26 Mapy řezů funkce příslušností s výstupním třídám v ose X3

Na Obr. 27 je promítnuta síť původních vah po učení bez učitele (40% krytí) do sítě výsledného stavu. V obrázku je patrný pohyb neuronů díky metody LVQ1. Toto doučení v průměru zlepší schopnost klasifikace sítě od 3 a více procent v případě dat s 200 vzory a 7 třídami.



Obr. 27 Znázornění pohybu sítě ze stavu před doučením do stavu po proběhnutí učení s LVQ1. Silně viditelná výsledná síť, slabě původní.

5.3.1 Srovnání kvality třídění

Nejlepším měřítkem kvality algoritmu je srovnání s jinými možnými metodami. Již bylo zmíněno, že během tvorby diplomové práce byly vytvořeny programy také Kohonenovy sítě a FKCN. Právě tento soubor algoritmů byl zvolen pro učení se na data ukázaná v kapitole 5.2. Pro učení byly nastaveny

Testovací soubor je složen ze tří datových souborů. První obsahuje 200 vzorů rozdělených do 7 tříd a nese název 0data.dat. Druhý soubor má 600 vzorů v 8 třídách v souboru 1data.dat. Poslední data byla generována do počtu 1000 vzorů a 9 tříd, která naleznete v souboru 2data.dat.

Nastavení sítě bylo jednotně 4x4 neurony. Parametr učení Kohonenovy a fuzzy-neuronové sítě se měnil v iteracích podle vztahu (33) pro počáteční hodnotu parametru učení $\mu_0 = 0,8$ s koeficientem exponenciály $k_\mu = 0,07$. Průběh parametru okolí byl modelován rovnicí (35). Počáteční hodnotou okolí bylo $\rho_0 = 3$ s krokem okolí $k_\rho = 2$ pro 200 vzorů a $k_\rho = 10$ pro 600 a 1000 vzorů. Maximum iterací bylo jednotně pro všechny 3

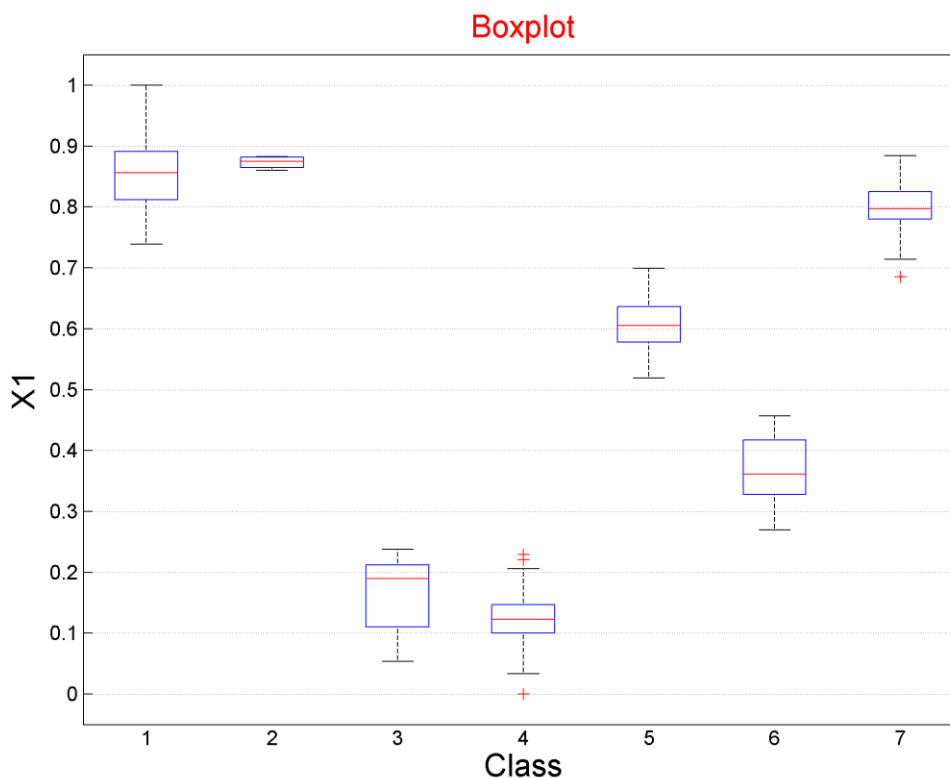
algoritmy určeno na 100 iterací. Od této hodnoty se odvíjel průběh fuzzy parametru u FKCN, který začínal na hodnotě $m_0 = 2$.

Pro všechny algoritmy byly při etapě doučení LVQ upraveny průběhy učení. U Kohonenovy a fuzzy-neuronové sítě počáteční parametr učení měl hodnotu $\mu_{LVQ} = 0,5$ a okolí bylo nastaveno rovno nule. FKCN začala doučování vždy s $m_0 = 1,5$.

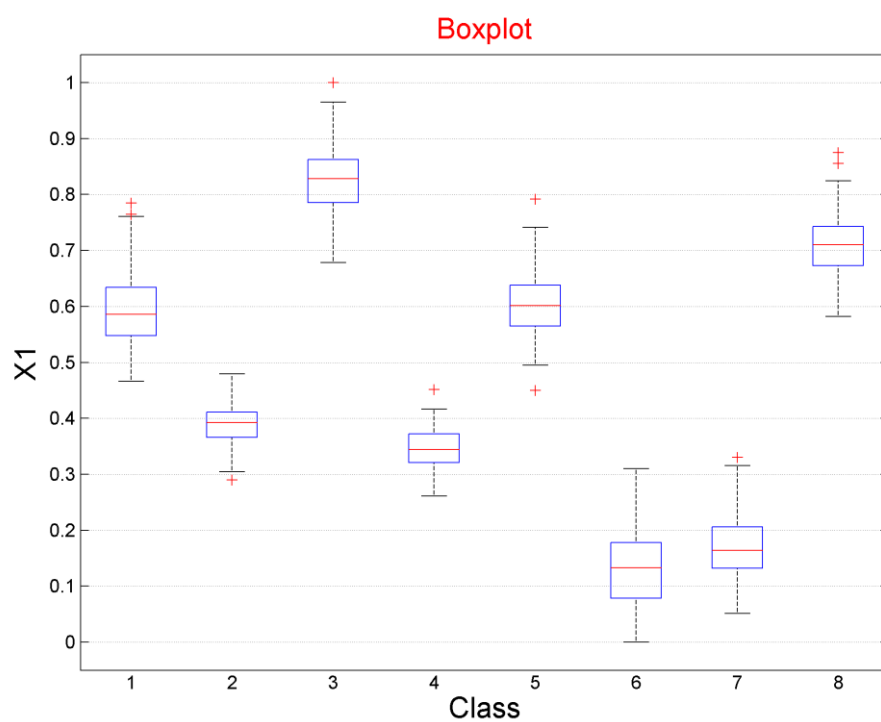
Protože výsledek byl závislý na pořadí předkládání tréninkových vzorů, muselo být pro všechny datové soubory provedeno opakování učení. Počet opakování byl zvolen 100. Po každém učení byla zapsána hodnota přesnosti třídění před i po doučení metodou LVQ.

Výsledná kvalita klasifikace je vykreslena v následujících grafech formou krabicových grafů. V krabicových grafech je červenou čarou u každého statistického souboru vyznačen medián (50% kvantil). Obdélník značí data od 25% kvantilu po 75% kvantil. Fousky v grafu značí rozmezí 5% až 95% kvantilu. Data mimo rozsah jsou znázorněna červenými křížky. Z tohoto grafického zobrazení lze odečíst nejen hodnoty kvantilů, ale můžeme pozorovat, zda jsou data významně odlišitelná.

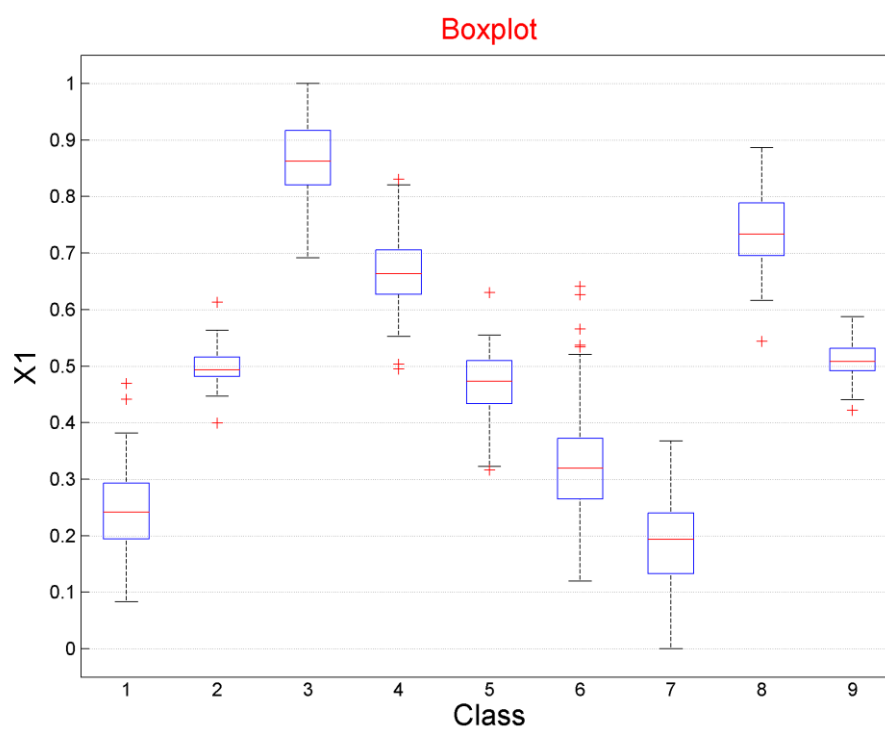
V grafech (viz Obr. 28, Obr. 29 a Obr. 30) byla vybrána první dimenze dat, která je nejlépe odlišitelná pro grafickou ukázkou. Celý soubor krabicových grafů vstupních dat je na příloženém CD.



Obr. 28 Data 200 vzorů 7 tříd - statistika v rozměru X1



Obr. 29 Data 600 vzorů 8 tříd - statistika v rozměru X1

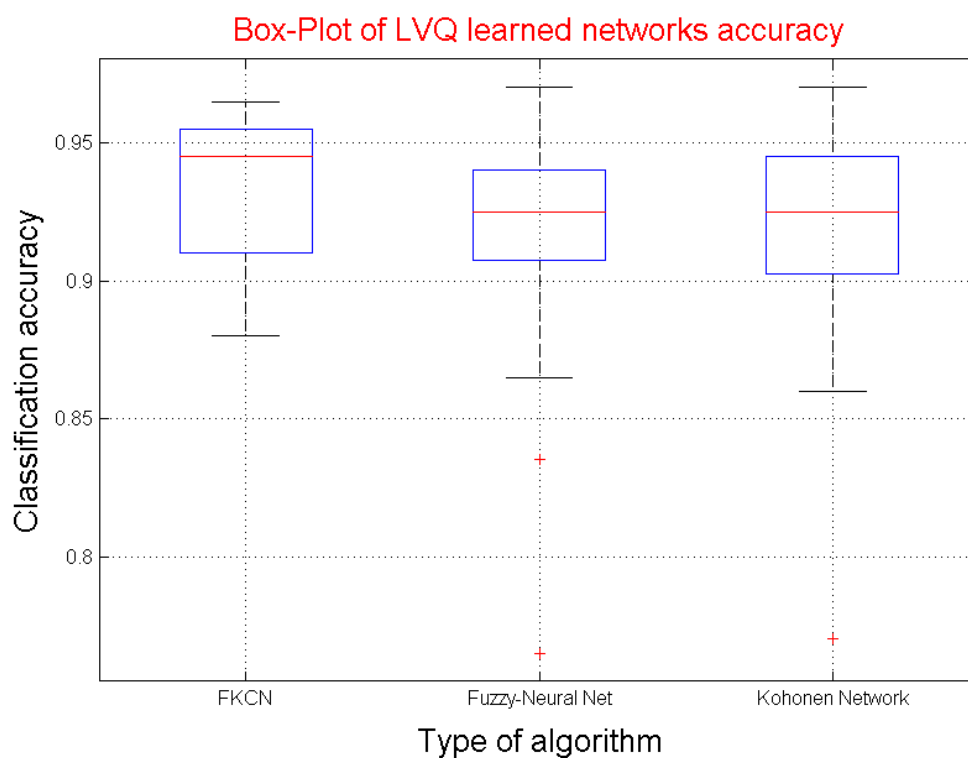


Obr. 30 Data 1000 vzorů 9 tříd - statistika v rozměru X1

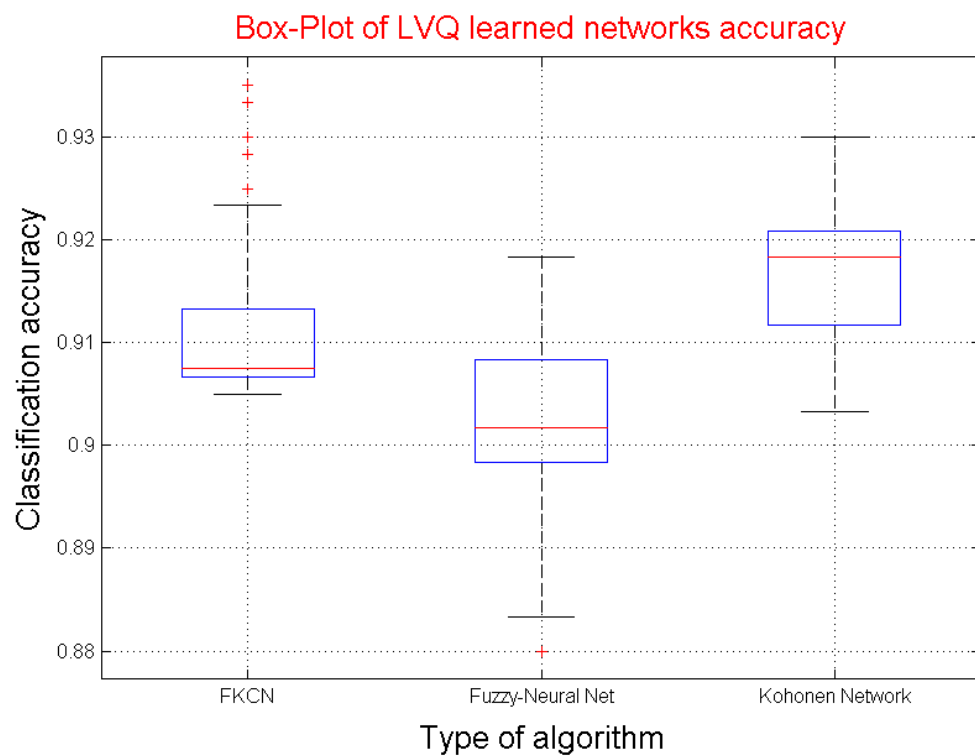
Krabicové grafy kvality třídění

Na následujících stránkách jsou vykresleny krabicové grafy hodnocení kvality třídění pro FKCN, fuzzy-neuronovou síť a Kohonenovu síť. Ve všech grafech fuzzy-neuronová síť zaostává za FKCN a Kohonenovou sítí. Ve zpětné analýze nastavení bych tvrdil, že tento viditelný rozdíl byl způsoben nastavením kroku okolí u větších datových souborů (600 a 1000 vzorů) na $k_p = 10$. To nejspíš vedlo k příliš mnoha iteracím, kdy byly stejně pozitivně adaptovány všechny základní fuzzy množiny a proto výsledné rozložení není tolik pozitivně nastaveno pro hodnocení vítězným reprezentantem.

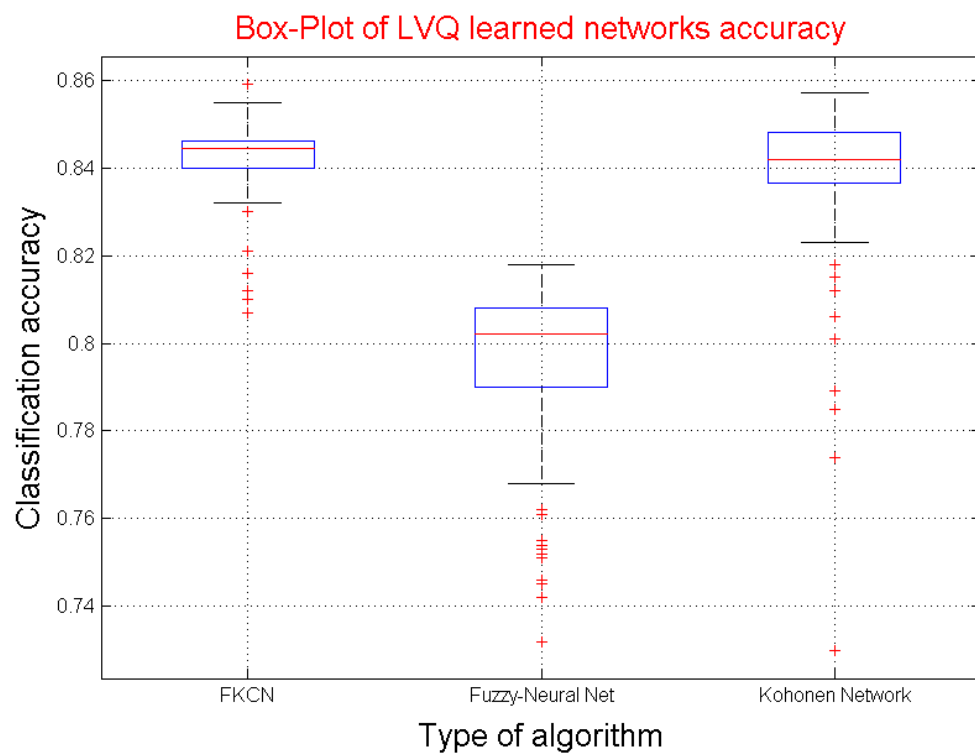
Rozptyl hodnocení fuzzy-neuronové sítě je nepatrně větší než u Kohonenovy sítě, ale na markantně užším intervalu hodnotí metoda FKCN. Grafy na Obr. 31, Obr. 32 a Obr. 33 jsou zpracovány z výsledného hodnocení po doučení LVQ metodou.



Obr. 31 Přesnost výsledné klasifikace pro 200 vzorů v datech

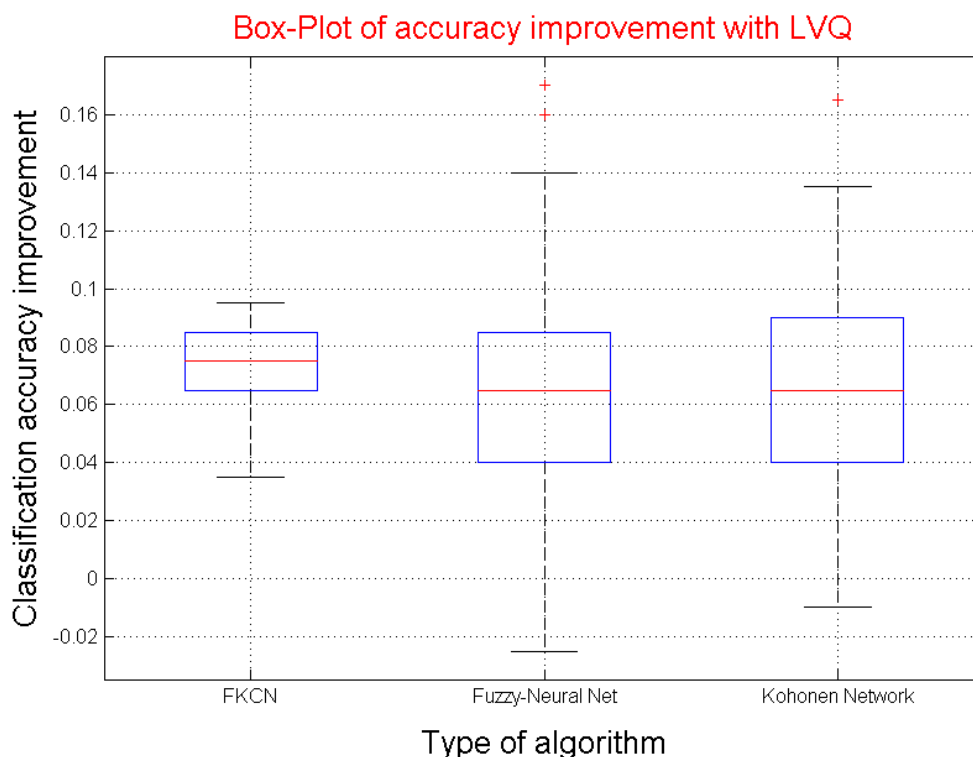


Obr. 32 Přesnost výsledné klasifikace pro 600 vzorů v datech



Obr. 33 Přesnost výsledné klasifikace pro 1000 vzorů v datech

V krabicových grafech na Obr. 31 a Obr. 32 dosahují všechny metody mediánem na přesnost klasifikace přes hodnotu 0,9 tedy 90% úspěšnost hodnocení. Na Obr. 32 medián hodnocení fuzzy-neuronové sítě zaostává maximálně o 2 % v kvalitě hodnocení. V případě 1000 vzorů a 9 tříd je ovšem již rozdíl v přesnosti klasifikace více než 4 %. Zhodnocení vlivu na výsledek v případě nasazení LVQ metody ukazuje jako náhled Obr. 34.

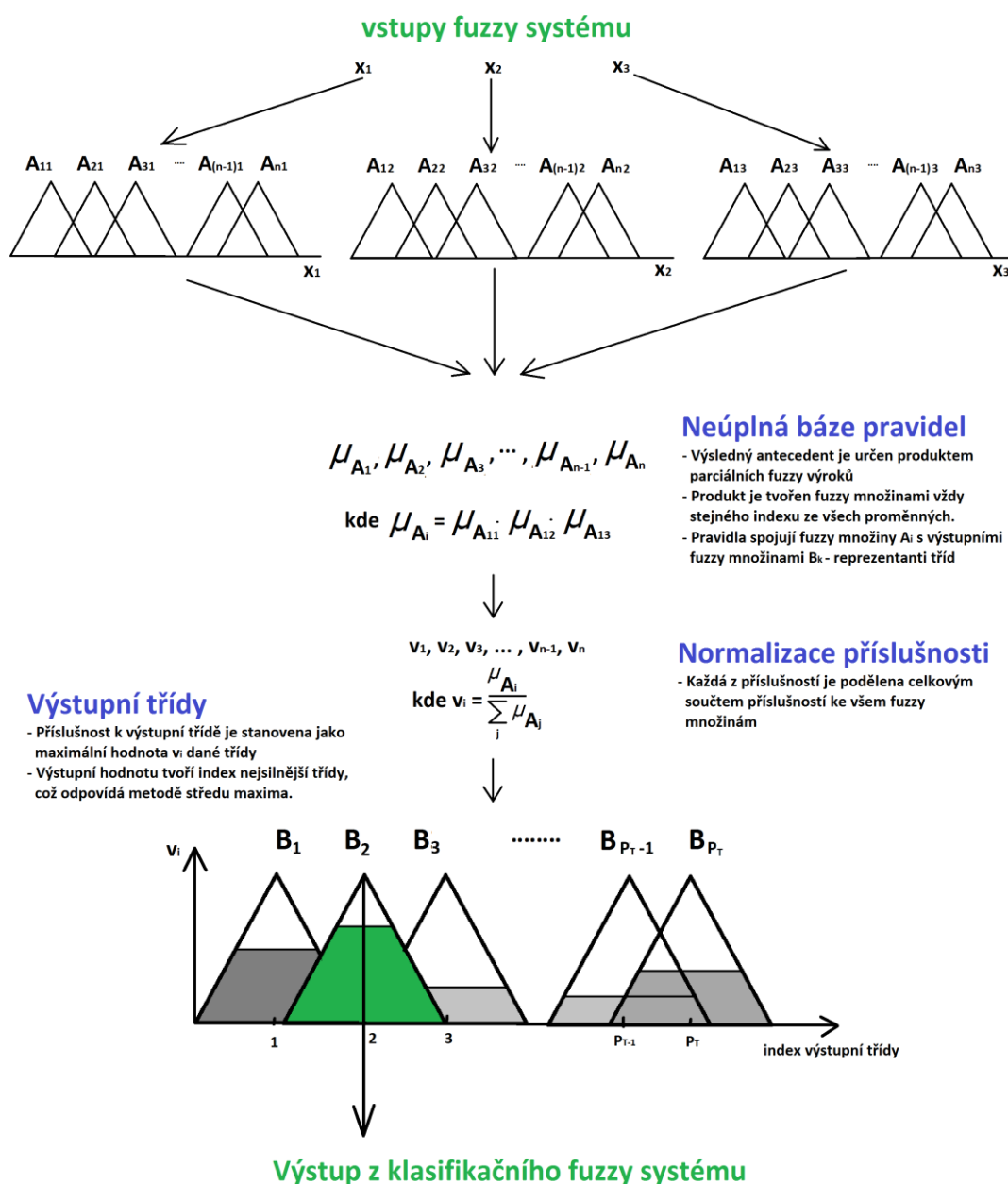


Obr. 34 Krabicový graf nárůstu přesnosti klasifikace po použití LVQ metody při učení na datový soubor 200 vzorů.

Obr. 34 ukazuje, že po použití LVQ metody doučení došlo dle 50% mediánu k asi 7% nárůstu přesnosti. Ovšem ve výjimečných případech toto zpřesnění může být až 17 %. Bohužel, protože dojde k pohybu středů fuzzy množin a to ne vždy správným směrem může u vyšších přesností klasifikace po učení bez učitele docházet po LVQ algoritmu k poklesu přesnosti, což značí záporné hodnoty v y-ose grafu.

5.4 Stavba fuzzy systému

Fuzzy systém je svým návrhem koncipován pouze jako klasifikátor. Z fuzzy-neuronové sítě pomocí učení získáme polohy středů a také ramena vstupních fuzzy množin. Parciální výroky antecedentů mají spojení v bázi pravidel vždy jen jeden z každé dimenze s jedním z každé z ostatních dimenzí vstupního prostoru. To znamená, že báze pravidel je neúplná. Vyhodnocení antecedentů je provedeno pomocí produktu. Příslušnost k výstupní fuzzy množině k . třídy je maximum z hodnot příslušností antecedentů, které patří danému konsekventu (výstupní třídě).



Obr. 35 Schéma klasifikačního fuzzy systému na bázi fuzzy-neuronové sítě

Defuzzifikace je zvolena ve formě Středu Maxima... tzn., že výstupní fuzzy množina s nejvyšší hodnotou příslušností postaví hodnotu svého středu základny na výstup (např. index třídy). Schéma systému ukazuje Obr. 35. Protože funkce plní funkci fuzzy systému je poměrně krátká, je vnořena v následující části textu. Pro její běh musí proběhnout učení fuzzy-neuronové sítě a předpokládáme znalost středů fuzzy množin včetně příslušnosti k třídě (matice *cw*), ramena základny (matice *L* a *R*). Funkce je uložena v podsložce programu */functions* a nese název *fuzzy_system*.

```
function [class, msg] = fuzzy_system(data, cw, L, R, fp,
rozmary_site, msg)
% Funkce reprezentuje Fuzzy Systém
% Vstupy: data - datový řádkový vektor o délce rovné počtu
%           dimenzí, na které se systém učil
%           cw - matice středů fuzzy množin včetně
%           informace o příslušnosti k výstupní třídě
%           L, R - matice parametrů základny fuzzy množin
%           fp - inicializovaná matice pro výpočet fuzzy
%           příslušnosti FBF
%           rozmary_site - vyska a sirka zavazbené sítě
%           msg - pole znaků předchozích zpráv
%
% Výstupy: class -třída předložených dat
%           msg - zpravodajský blok doplněný o informaci
%           z funkce.

vyska = rozmary_site(1);
sirka = rozmary_site(2);

% Obrácení vektoru dat
if(size(data,2)<size(data,1))
    data = data';
end

% Ověření rozměru dat
if(~(size(data,2)==size(cw,3)-1))
    disp('Data nemají odpovídající počet dimenzí'); return;
end

% Výpočet fuzzy příslušnosti dat ke každé FBF v bázi dat
for i = 1:vyska
    for j = 1:sirka
        fp(i,j) = fcePrislusnostiAprod
(cw(i,j,:),L(i,j,:),R(i,j,:),data);
    end
end
```

```

% Normalizace hodnot příslušnosti
sum_fp = sum(fp(:));
for i = 1:vyska
    for j = 1:sirka
        fp(i,j) = fp(i,j)/sum_fp;
    end
end

% Výběr nejlepšího zástupce z výstupní tříd
[r,s] = find(fp == max(max(fp)),1);
class = cw(r,s,end);

clc; msg = [msg, strcat('Předložená data jsou klasifikována
do třídy :', num2str(class)), ...
sprintf('\n')]; disp(msg);

```

Výpočet výstupní hodnoty jsem provedl zjednodušenou formou bez přímé tvorby výstupních fuzzy množin.

6 ZÁVĚR

V práci bylo provedeno krátké shrnutí základů fuzzy logiky a vymezil jsem parametry, které jsou často využívány k adaptaci fuzzy systémů. Byla provedena rešerše metod spojení fuzzy systému a fuzzy logiky s neuronovými sítěmi a zhodnocena možnost adaptace parametrů fuzzy-systému. V rámci rešerše jsou uváděny také zdroje studentských prací, bakalářských nebo diplomových, které se danými algoritmy zabývají s uplatněním na praktických problémech jako například prohledávání prostoru a tvorba map mobilních robotů nebo klasifikace spánkových EEG.

Z možných algoritmů jsem vybral fuzzy-neuronovou síť na bázi Kohonenovy samoorganizující mapy. Mimo jiné byla v algoritmu zahrnuta adaptace LVQ, která po naučení Kohonenovy sítě, provede doučení s učitelem ze znalosti třídy příslušnosti grandmother cell tréninkových dat. V rámci práce se zvolenou fuzzy-neuronovou sítí byly doplněny algoritmy Kohonenovy sítě a Fuzzy Kohonenovy Clusterovací Sítě. Byly zhodnoceny jejich přednosti a v páté kapitole jsem provedl rozbor ze simulace klasifikace těmito metodami. Formou krabicových grafů jsou vyneseny hodnoty přesnosti klasifikace pro 100 opakování jednotlivých algoritmů.

Z dat získaných opakovaným učením na předložené vzory a to v datových souborech: 200 vzorů v 7 třídách, 600 vzorů v 8 třídách a 1000 vzorů v 9 třídách vyplývá následující. Rozptyl hodnot přesnosti od mediánu vlivem pořadí předložených vzorů vykazuje nejmenší síť FKCN. FKCN dosahovala celkově dobrých výsledků pro všechny datové soubory. Kohonenova síť byť s vyšším rozptylem hodnot výsledné klasifikační přesnosti se mediánem pohybovala ve všech případech velmi blízko FKCN. Bohužel patrně nevhodným nastavením parametrů zvolená fuzzy-neuronová síť za oběma předchozími zaostávala, i přesto v případě dat 200 vzorů i 600 vzorů klasifikovala 50% kvantilem přes 90 % vzorů úspěšně. Dále byl ukázán vliv modifikace LVQ1, který v Obr. 34 vykazuje medián zlepšení klasifikační schopnosti o 7 % a maximálně potom 17 %.

Motivací pro tvorbu fuzzy systému na fuzzy-neuronové síti s Kohonenovým učením je především snadné sestavení báze fuzzy pravidel, která jsou již prakticky vyjádřena v naučené síti formou středu a ramen základy fuzzy množin a znalosti výstupní třídy. Tento algoritmus pro svoji poměrnou rychlost a značnou kompresi dat díky reprezentaci shluků při učení je vhodnou metodou pro další nasazení.

Dalšími možnými úpravami by bylo spojení algoritmu s netriviálními fuzzy množinami na výstupu a pomocí iterativních gradientních metod umožnit modelování libovolné funkce na základě znalosti reprezentantů shluků. Vhodným krokem do budoucna by bylo analogické vytvoření fuzzy systému na základě FKCN.

7 LITERATURA

- [1] AL-JARRAH, Omar M. – BANI-MELHEM, Omar Q.: Building maps for mobile robot navigation using fuzzy classification of ultrasonic range, in: *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, IOS Press, Amsterdam. Volume 11, Issue 3,4. December 2001. Pages 171 – 184. ISSN: 1064-1246
- [2] De ALMEIDA, Carlos W. D. – De SOUZA, Renata M. C. R. – CANDEIAS, Ana L. B.: Fuzzy Kohonen clustering networks for interval data, in: *Neurocomputing*, Volume 99, 1 January 2013, Pages 65-75, ISSN 0925-2312.
- [3] ATMACA, H. – BULUT, M. – DEMIR, D.: Histogram Based Fuzzy Kohonen Clustering Network for Image Segmentation, in: *Proceedings of International Conference on Image Processing*, IEEE 1996. Volume 1. Pages 951 - 954 vol. 2. Lausanne 1996. ISBN 0-7803-3259-8.
- [4] BEZDEK, J.C. – TSAO E.C.-K. – PAL, N.R.: Fuzzy Kohonen clustering networks, in: *Proceedings of the First IEEE Conference on Fuzzy Systems*, SanDiego, USA, 1992.
- [5] BURÝ, A.: *Teorie systémů a řízení*, skripta, VŠB – Technická univerzita Ostrava. Ostrava, 2007. 62 s. [cit. 7. 1. 2015], Dostupné na URL: <<http://homen.vsb.cz/~bur50/TAR07.pdf>>
- [6] FIC, M.: *Kohonenova síť*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2012. 59 s. Vedoucí bakalářské práce byl doc. Ing. Václav Jirsík, CSc.
- [7] FULLÉR, Robert.: *Neural Fuzzy Systems*. Åbo Akademis tryckeri, Åbo, ESF Series A:443, 1995., 249 s. ISBN 951-650-624-0, ISSN 0358-5654.
- [8] JABBAR, N. I. – AHSON, S. I.: Modified Fuzzy Kohonen Clustering Network for Image Segmentation, in: *International Conference on Financial Theory and Engineering*. Dubai, United Arab Emirates 2010.
- [9] JINHUI Fan – SONGMIN Jia – XIUZHI Li: The Application of Fuzzy Kohonen Clustering Network for Intelligent Wheelchair Motion Control, Proceeding of the IEEE, *International Conference on Robotics and Biomimetics (ROBIO)*. Shenzhen, China, December 2013. p. 1995-2000.

- [10] JURA, P.: *Základy fuzzy logiky pro řízení a modelování*, skripta z předmětu Fuzzy systémy, FEKT VUT v Brně, Brno, 2014. 120 s.
- [11] KOSKO, B.: Fuzzy systems as universal aproximators. Proc. IEEE Int. Conf. on Fuzzy Systems, San Diego 1992, pp.1153-1162.
- [12] KOVACS, T.: *Genetic Fuzzy Systems*, [online]. 12. 3. 2011, [cit. 24. 1. 2015], URL: < <http://www.cs.bris.ac.uk/~kovacs/publications/gbml-survey/html-version/node16.html>>
- [13] MACHEK, O. *Fuzzy modely map pro pohyb mobilních robotů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 60s. Vedoucí diplomové práce byl prof. Ing. Pavel Jura, CSc.
- [14] OLLÉ, T. *Fuzzy neural networks for pattern classification*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav radioelektroniky, 2012. 50 s., 8 s. příloh. Diplomová práce. Vedoucí práce: Ing. Jitka Svobodová
- [15] RAHMAT, Basuki – JOELIANTO, Endra.: Adaptive Neuro Fuzzy Inference System(ANFIS) with Error Backpropagation Algorithm using Mapping Function, in *International Journal of Artificial Intelligence*. Autumn 2008, Vol. 1, Number A08,s. 3-8. ISSN 0974-0635.
- [16] RONZHINA, M.: *Klasifikace mikrosprávku analýzou EEG: diplomová práce*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 69 s. Vedoucí diplomové práce Ing. Vratislav Čmiel.
- [17] SVALINA, I. A KOL.: An adaptive network-based fuzzy inference system (ANFIS) for the forecasting: The case of close price indices, in *Expert Systems with Applications*, 2013. Volume 40, Issue 15, Pages 6055-6063, ISSN 0957-4174
- [18] ŠÍMA, J., NERUDA, R.: *Teoretické otázky neuronových sítí*. Praha: Matfyzpress, 1996. 390 s. ISBN 80-85863-18-9.
- [19] TUČKOVÁ, J.: *Umělé neuronové sítě pomáhají lékařům FN Motol*. [online]. Datum publikace 2008-03-10 [cit. 2012-05-20]. Dostupné z URL: <<http://www.3pod.cz/254-Umele-neuronove-site-pomahaji-lekarum-FNMotol.html>>.
- [20] ZITTA, R. a PALÁTOVÁ, M.: *Metody pro generování fuzzy pravidel z dat*, Vysoká škola chemicko-technologická Praha, Ústav počítačové a řídicí techniky 2004.

Seznam symbolů

α	parametr fuzzy množiny
β	parametr fuzzy množiny
γ	parametr fuzzy množiny
Δ	matice prostých osových vzdáleností vah a dat
ε	mezní hodnota chyby, nebo krok parametru učení
μ_F	hodnota funkce příslušnosti
μ_{LVQ}	parametr doučení (metoda LVQ)
μ_0	parametr učení
ρ_0	okolí neuronu
θ	parametr fuzzy učení
A	fuzzy-množina
B	fuzzy-množina
E	chyba sítě
L	matice levých ramen fuzzy množin
R	matice pravých ramen fuzzy množin
S	suma prostých osových vzdáleností
V	fuzzy příslušnost pro doučení ramen L , R
c	index nejbližšího neuronu
d	žádaná hodnota výstupu
k_μ	koeficient exponenciály
$k\rho$	krok okolí
m	fuzzy parametr
m_0	počáteční hodnota fuzzy parametru
n	počet neuronů (rozměr vstupních dat)
p	počet tréninkových vzorů
s	index tréninkového vzoru.
t	index iterace
u	fuzzy příslušnost
v	třída neuronu
y	výstup sítě/ výstupní proměnný systému
x	vstup
\mathbf{x}	vstupní vektor, tréninkový vzor
w	váha
\mathbf{w}	váhový vektor

Seznam zkratek

AEN	Neuronová síť pro Evaluaci Akčního stavu
ANFIS	Adaptivní Neuro-Fuzzy Inferenční Systém
ARIC	Aproximativní Inteligentní regulátor na bázi usuzování
ASN	Neuronová síť – Akční Selektor
CAM	Metoda Průměru Středů
CoG	Metoda Centra Gravitace
CoS	Metoda Středu Součtů
EEG	Elektro EncefaloGram
FBF	Fuzzy Bázová Funkce
FCM	Fuzzy C-Means
FEKT	Fakulta Elektrotechniky a Komunikačních Technologií
FIS	Fuzzy Inferenční Systém
FKCN	Fuzzy Kohonenova Clusterovací síť (Network)
FNN	Fuzzy Neuronová síť (Network)
FoM	Metoda Prvního Maxima
KCN	Kohonenova clusterovací síť (Network)
KNN	Kohonenova Neuronová síť (Network)
LVQ	Learning Vector Quantization (učicí vektorová kvantizace)
MATLAB	Matematická Laboratoř – Software od MathWorks
MoM	Metoda středu maxima
RBF	Radiální Bazická Funkce
VUT	Vysoké Učení Technické v Brně

Seznam příloh

CD obsahuje:

- Elektronickou verzi diplomové práce
- Hlavní skript *a150728_Networks_TestScript.m*
- Obrazové podklady z výpočtu se složce */pictures*
- Funkce jsou uloženy ve složce */functions*